A Blockchain-Based Tool for Enhancing Digital Evidence Integrity in the Chain of Custody

Laban Machuki Nyarera

119887

Submitted in Partial Fulfilment of the Requirements for the Degree of Master of Science in Information Systems Security at Strathmore University

School of Computing & Engineering Sciences

Strathmore University

Nairobi, Kenya

June, 2025

Declaration and Approval

Declaration

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the dissertation contains no material previously published or written by another person except where due reference is made in the dissertation itself.

Student's Name: Laban Machuki Nyarera

Sign: 

Date: 21st May 2025

Approval

This dissertation of Laban Machuki Nyarera was reviewed and approved for examination by the following:

Dr. Humphrey Njogu,

Senior Lecturer, School of Computing & Engineering Sciences,

Strathmore University

Dr Julius Butime,

Dean, School of Computing & Engineering Sciences,

Strathmore University

Prof. Bernard Shibwabo,

Director of Graduate Studies,

Strathmore University

Abstract

Digital evidence is crucial in cybercrime investigations and is vital in identification and prosecution. However, preserving its integrity throughout the chain of custody is crucial for its reliability and ultimate admissibility. Unlike physical evidence, digital evidence is highly volatile, making it susceptible to duplication and alteration. The distinctive attributes of digital evidence pose significant challenges to traditional evidence management methods, which are susceptible to manipulation, unlawful access, and data corruption. This research addresses these gaps by creating a blockchain-based digital forensics chain of custody system. Blockchain technology provides an immutable, transparent, and decentralized ledger, ensuring that digital evidence remains tamper-proof and traceable. Agile methodology was used to implement the solution to accommodate changing project requirements. The testing was conducted using a simulated investigative process, examining blockchain transactions associated with various stakeholders along the chain of custody.

The key findings indicate that the blockchain-based tool ensures evidence integrity with secure, verifiable audit trails. Validation confirmed 100% functional accuracy in automated timestamping. Performance tests demonstrated high uptime. API response times were within acceptable limits, with slight delays in case creation due to blockchain latency. Security testing verified tamper-proof integrity with zero unauthorized breaches. The validation confirmed the system's capability to maintain a chain of custody for digital evidence, ensuring data integrity and security as blockchain transactions remained immutable and traceable.

Keywords: Digital Forensics, Digital evidence, Chain of custody, Blockchain.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI - Artificial Intelligence

API - Application Programming Interface

DSR - Design Science Research

IoT - Internet of Things

Acknowledgement

I would like to express my gratitude to my supervisor Dr. Humphrey Njogu for his invaluable support throughout this research. I also want to acknowledge my family for their constant encouragement and backing during this journey.

Chapter 1: Introduction

1.1 Background

The continuous advancement and adoption of technology has raised the risk of cyber related attacks with cybercriminals exploiting the speed, convenience, and anonymity of the Internet technologies to commit crimes. According to the PricewaterhouseCoopers (PWC) Global Economic Crime and Fraud Survey report (2022), cybercrime continues to be one of the biggest potential threats to organizations. This rapid rise of cybercrime has led to the growing importance of digital evidence for provenance of persons linked with cybercrimes (Lone & Mir, 2019).

According to Shah et al. (2017) evidence is key in solving any crime. Yunianto et al. (2019) further identifies digital evidence as the basis of any digital forensic process with credible digital evidence maintained for law and court process. Digital evidence is therefore important as it not only links persons with cybercrime but can also be used for possible prosecution. Digital evidence may become inadmissible if its integrity cannot be proved (Shah et al., 2017). A chain of custody is susceptible to compromise if documentation is not properly maintained and preserved during the lifecycle of digital evidence. The integrity of the chain of custody from the time of collection till the prosecution is therefore key to ensure evidence is admissible and acceptable.

According to Prayudi (2015), the unique characteristics of digital evidence make its handling complicated and complex considering the exchanges of evidence between interested parties. Digital evidence characteristics such as ease of transmission, vulnerability to tampering, time sensitivity and the ability to cross legal jurisdictions make digital evidence fragile and volatile compared to physical evidence (Lone & Mir, 2019).

The fragile nature of digital evidence makes it susceptible to integrity violations as it traverses the various tiers of the hierarchy throughout a digital forensic inquiry. This ultimately leads to the integrity challenge of ensuring that digital evidence presented is complete, unaltered and that any alterations and access is only done by authorized parties. It is against this backdrop that chain of custody is identified as a way of preserving the integrity of digital evidence.

Giova (2011) defines chain of custody as a process used to maintain and document the chronological history of handling digital evidence. According to Prayudi and SN (2015), a chain of custody documents where, when, why, who, how digital evidence is used and covers the whole investigative process. Evidence admissibility is better associated with the existence of a solid chain of custody that enables verification of the authenticity of evidence and provides assurance against tampering (Yeboah-Ofori & Brown, 2020). Preservation of digital evidence can therefore be realized by a chain of custody to ensure the integrity of the evidence.

Blockchain technology is one of the promising technologies to address integrity issues of digital evidence due to its inherent properties that can be leveraged to ensure integrity of digital evidence is maintained. Blockchain is a series of connected data structures called blocks, which contains or tracks everything that happens on some distributed systems on a peer to peer to network (Nakamoto,2008). An append-only system is created when each block is connected to and dependent upon the prior block, creating a permanent and irreversible history that any participant can use as a real-time audit trail to confirm the accuracy of records by just looking at the data itself. Its capability of enabling comprehensive view of transactions back to origination therefore provides enormous capability for implementing a digital evidence chain of custody. Blockchain inherently ensures transparency, authenticity, security, and auditability, making it ideal for maintaining and tracing the chain of custody in forensic applications (Lone & Mir, 2019). The focus of this study is to create a blockchain based solution that safeguards the integrity of the forensics chain of custody.

1.2 Problem Statement

The increase in cybercrime has elevated importance of digital evidence in digital forensic investigation, as it is key to the identification and prosecution of cybercrime perpetrators. However, for the evidence to be admissible in court a chain of custody documenting how evidence was gathered, transported, analysed, and presented thereby proving that evidence has not been altered or changed must be kept. Maintaining the integrity of digital evidence with regards to chain of custody is however challenging due the unique characteristics of digital evidence and the frequent exchanges as it traverses several hierarchical levels during a digital forensic examination. Unlike physical evidence, digital evidence is vulnerable to tampering as it can be easily copied, altered, damaged, or destroyed making it susceptible to malicious alteration or accidental changes and ultimately inadmissible. Documentation of digital evidence is also complicated by its ease of access and duplication. Another challenge is the security of chain of custody documentation, considering that evidence can be transferred between parties. Despite the development of chain of custody solutions, there are still integrity concerns primarily due to the reliance on manual processes which are vulnerable to human error, data entry mistakes, and tampering. It is against this backdrop that a robust automated system with the ability to provide an audit trail and maintain the integrity of the evidence itself is proposed. This study addresses these challenges by designing and developing a blockchain-based chain of custody solution to enhance digital evidence integrity in the chain of custody

1.3 Research Objectives.

General Objective

The aim of this study is to develop a chain of custody solution that will leverage blockchain technology to improve the integrity of digital evidence along the digital chain of custody.

Specific objectives

The specific objectives of the study are.

1. To identify the common challenges that affect the integrity of evidence along the chain custody.
2. To review the existing solutions for preserving integrity of evidence along the chain custody.
3. To design, develop and test a digital forensic chain of custody platform that leverages blockchain technology.
4. To validate the effectiveness of the proposed prototype.

1.4 Research Questions

The research shall attempt to answer the following questions:

1. What are common challenges that affect the integrity of digital evidence along the chain custody?
2. What are the existing solutions for preserving integrity of evidence along the chain custody?
3. How can a blockchain based chain of custody solution be designed, developed, and tested?
4. How can the proposed digital chain of custody solution be validated?

1.5 Justification

The digital evidence life cycle is becoming more intricate with each phase increasing the risk of breaches that could compromise its integrity. This coupled with the unique characteristics of digital evidence presents difficulties in the documentation of digital evidence. A system that maintains the integrity of evidence along the chain of custody will enhance the admissibility of digital evidence in courts and increase societal trust in its credibility and reliability.

1.6 Scope and Limitations

This dissertation focuses on digital evidence and the chain of custody in digital forensic investigations. The research aims to enhance the integrity of digital evidence during its lifecycle, specifically addressing the stages of collection, documentation, storage, and transfer. The solution does not cover the identification of evidence, evidence examination, and evidence analysis.

Chapter 2: Literature Review

2.1 Introduction

This section provides literature background on digital forensics, digital evidence, chain of custody challenges. It examines various technologies employed in digital chain of custody and provides an in-depth analysis of Blockchain technology. The section also reviews the existing solutions for preserving integrity of digital evidence.

2.2 Theoretical Literature

2.2.1 Locard's Exchange Principle

Locard's Exchange Principle states that "every contact leaves a trace" (Locard, 1930). Originally applied to physical forensics, the principle in digital forensics emphasizes the preservation and integrity of data traces during digital interactions (Spichiger & Adelstein, 2025). In digital forensics, Locard's principle highlights the need to identify, preserve, and analyse digital traces to reconstruct events and maintain the integrity of forensic investigations (Malik & Sharma, 2023). Locard's principle is relevant to the investigation process and chain of custody as it reinforces the importance of preserving digital evidence to ensure admissibility.

2.2.2 Provenance Theory

The provenance theory builds on Locard's concept of traceability and focuses on the documentation of an artifact's origin, history, and custody (Haque & Atkison, 2018). Akbarfam et al. (2024) define provenance as the process of tracing and authenticating the origin, custody, and history of any data artifact throughout the entire investigative process. In digital forensics, accurate provenance is essential to ensuring the credibility and integrity of digital evidence (Akbarfam et al., 2023). The documentation of evidence enhances authenticity, reliability, and accountability, making it a crucial component of digital evidence integrity. Provenance supports the chain of custody by establishing a verifiable audit trail that records when data was accessed, by whom, and how it was modified, thereby mitigating risks of tampering, loss, or unauthorized access (Pourvahab & Ekbatanifard, 2019)

2.2.3 Evidence Lifecycle Theory

The evidence lifecycle theory, rooted in forensic science principles and records management standards such as ISO 15489-1:2016 (International Organization for Standardization [ISO], 2016), offers a structured framework for managing evidence from its inception to its final disposition. According to Casey (2011), the evidence lifecycle theory encompasses five stages: collection, preservation, analysis, presentation, and disposal, ensuring systematic management of evidence and establishing an audit trail. In digital investigations, maintaining lifecycle integrity is critical, as digital evidence is highly volatile and susceptible to corruption or loss if not properly handled (Garfinkel, 2010). By systematically managing evidence through defined lifecycle stages, investigators ensure compliance with legal, forensic, and procedural standards (Carrier & Spafford, 2004). The lifecycle approach also strengthens the chain of custody by providing a structured mechanism for tracking and controlling access to digital evidence, thereby ensuring their integrity and admissibility in court.

2.2.2 Digital Forensic Fundamentals

2.2.3.1 Overview

The National Institute of Standards and Technology (NIST) in Special Publication 800-86 defines digital forensics as "the application of science to the identification, collection, examination, and analysis of data, while preserving the integrity of the information and maintaining a strict chain of custody for the data." Baskar (2025), on the other hand, defines digital forensics as "the field of forensic science that focuses on identifying, preserving, analysing, and presenting digital evidence in a legally admissible manner." Digital forensics provides an understanding of previous breaches and is therefore key in disclosure of cybercrime as it can be used to investigate incidents and gather admissible digital forensic evidence that can aid judicial review.

2.2.3.2 Digital Evidence

Stoykova (2021) defines digital evidence as any information processed by electronic means that supports or refutes a hypothesis about the state of digital artefacts or events, with potential relevance and probative value in a criminal investigation. In this research, digital evidence is therefore considered as data of investigative value, stored on or transmitted by a digital device, that may help establish a link between a crime and its perpetrator. The growing digital network and an upsurge of cybercrimes have therefore made digital evidence important, as it is used to prove facts or to convict personnel involved in cybercrimes (Lone & Mir 2019).

Unlike physical evidence, documentation of digital evidence is complicated by its unique characteristics such as ease of access, duplication, and transfer between parties (Alruwaili, 2021). Digital evidence is also vulnerable to tampering as it can be easily copied, altered, damaged, or destroyed making it susceptible to malicious alteration or accidental changes. The unique characteristics of digital evidence present distinct challenges regarding chain of custody.

2.2.3.3 Chain of custody

Kebande and Venter (2021) define the chain of custody as a chronological and legally admissible record detailing the seizure, control, transfer, analysis, and disposition of both physical and digital evidence. In contrast, Alqahtany and Syed (2024) conceptualize the chain of custody as a legal process that ensures the proper documentation and preservation of evidence as it is transferred among stakeholders during investigations or judicial proceedings. In digital forensic investigations, chain of custody is a crucial as it documents all the specific information related to digital evidence as it progresses through different hierarchical levels starting with the first point of contact and advancing to authorities responsible for conducting cybercrime investigations (Lone & Mir, 2019).

For the evidence to be accepted by the court as valid, the chain of custody for digital evidence must be kept, or it must be known who exactly, when, and where came into contact with the evidence at each stage of the investigation (J. Cosic & Z. Cosic, 2012)

In the court of law, if the chain of custody is weak, it could result in the digital evidence being deemed inadmissible. The chain of custody is vital in forensic investigations as it enables forensic investigators to track where, when, how, and by whom the evidence was discovered, collected, and handled (Cosic et al., 2021)

The life cycle of digital evidence is very complex, and at each stage there are potential risks that can violate the chain of custody. Prayudi and Azhari (2015) highlight the primary challenge as the increased complexity of digital evidence documentation due to the unique digital evidence characteristics coupled with user mobility trends allowing exploration and analysis anywhere and anytime. Another challenge is the security of chain of custody documentation, considering that evidence can move from one party to another.

2.2.3.4 Challenges in Digital Forensics

Digital forensics faces several challenges in maintaining evidence integrity, efficiency, and legal admissibility. The enhancement of forensic tools and legal frameworks is therefore essential to ensuring the security and reliability of evidence (Igonor et al., 2025). Karie and Venter (2015) categorize digital forensic challenges into four main groups: technical, legal, personnel-related, and operational. Igonor et al. (2025) further analyse the challenges identified by Karie and Venter (2015) and align them with specific requirements to ensure comprehensive coverage. Table 2.1 below presents Igonor et al.'s (2025) classification.

Table 2.1: Requirements and challenges in digital forensics (Igonor et al.2025)

Category	Challenges	Description
Technical Requirements	High Volume of Data	Efficiently managing and processing voluminous datasets. Data decentralization, accessibility, duplication and management during investigations. Variation of data along with data sources.
	Emerging Technologies	Challenges driven by adoption of emerging digital gadgets and technology like AI, IoT, and cloud.
	Digital Evidence Security	Maintaining integrity, preventing tampering, ensuring confidentiality, and guaranteeing availability of digital evidence.
	Forensics Process Automation	Manual processes in forensic investigations. Automate tasks for efficiency
Legal Systems Requirements	Jurisdiction	Legal complexities stemming from cross-border data storage, exchange, and access, along with varied legal frameworks
	Admissibility of Digital Forensic Methods and Tools	Ensuring the legal acceptance of forensic techniques.
	Privacy and Ethical Concerns	Striking a balance between privacy rights and investigative demands
	Chain of Custody	Ensure the integrity, reliability, and documentation of evidence from collection to presentation, while also managing access control and preventing tampering.

2.2.3.1 Legal and Regulatory Context

The legal and regulatory framework surrounding digital evidence and the chain of custody is crucial for maintaining its integrity and admissibility in legal proceedings. Admissibility standards require digital evidence to be relevant and reliable, with strict adherence to protocols that prevent tampering or alteration. In Kenya, the Evidence Act (Cap 80) governs digital evidence, with Section 78A recognizing electronic evidence as admissible and Section 106B mandating certification for authentication before it is presented in court. The Computer Misuse and Cybercrimes Act, 2018, establishes procedures for handling cyber offenses and collecting digital evidence, while the Data Protection Act, 2019, safeguards personal data in legal proceedings, ensuring lawful collection and protection against misuse. Internationally, ISO/IEC 27037:2012 provides guidelines for identifying, collecting, acquiring, and preserving digital evidence.

2.2.4 Technologies for enhancing integrity of digital evidence.

2.2.4.1 Digital Signatures

Digital signatures are mathematical methods or algorithms employed to verify the validity and integrity of information or messages, such emails, credit card transactions, or digital documents (Alenezi et al., 2020). Digital signatures function as electronic fingerprints that uniquely identify users and safeguard data. A digital signature guarantees that a communication or document remains unaltered from the moment of signing. The process involves hashing the document or message followed by encrypting it with the sender's private key. Digital signatures use Public Key Infrastructure (PKI) to strengthen security. Digital signatures can be used to verify the integrity of a file or a message (Fang et al., 2020).

2.2.4.2 Encryption

Encryption, the technique of encoding information to prevent unauthorized access and guarantee data integrity and secrecy (Stallings, 2017). Data encryption is the technique of protecting messages by transforming them into hidden texts while decryption is the reverse process of recovering original texts from hidden texts, (Alenezi et al., 2022).

Encryption can be used to enhance the integrity of digital evidence by ensuring only authorized individuals can access it and that any modifications to the evidence are detectable.

2.2.4.3 Hashing

Cryptographic hash function is a function that converts a message of any length to data of fixed length. According to Ali et al (2022), hashing is a cryptographic method for determining an entity's unique representation. Hashing algorithms play a crucial role in ensuring the integrity of evidence with most tools relying on hashing functions such as SHA-1 and MD5 (Rasjid et al., 2017).

2.2.4.4 Extensible Markup Language (XML)

Extensible Markup Language (XML) is a markup language and file format for storing, transmitting, and reconstructing arbitrary data. XML can be used to save evidence information with hash value to determine if a file has been tampered with or deleted. The XML concept can be used to preserve the integrity of the hash value of evidence files so that the evidence is well documented and acceptable in court.

2.2.4.5 The Emerging Role of Blockchain in Digital Forensics

The concept of blockchain was originally proposed by Nakamoto in 2008 as an underlying construct of Bitcoin. Bezuidenhout et al. (2023) define blockchain as a tamper-proof distributed ledger with cryptographically linked blocks that are updated through decentralized consensus models. By its very design, blockchain ensures transparency, authenticity, security, and auditability making it an ideal solution for preserving the integrity of digital evidence.

2.2.5 Understanding Blockchain Technology

A blockchain is a secure and decentralized digital ledger composed of linked blocks of data. Each block contains a header and a body. The header includes key information such as the version number, timestamp, Merkle root (which summarizes all transactions in the block), the block's own hash, and the hash of the previous block. The body holds the list of transactions. This structure creates a cryptographic chain between blocks, making the

data tamper resistant. Once information is added to a block, it cannot be altered. Any changes or updates are recorded as new transactions in subsequent blocks. This ensures the integrity, immutability, and traceability of data throughout the blockchain.

As illustrated in Figure 2.1, each block in the blockchain contains a header with a version number, timestamp, and a cryptographic hash of the previous block, forming a secure and immutable chain (Chen et al., 2023).

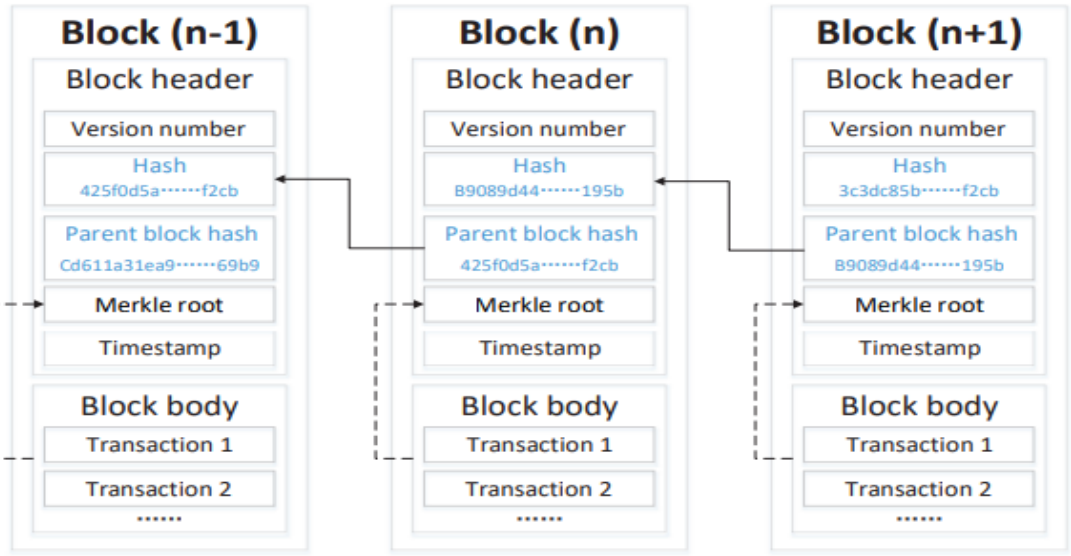


Figure 2.1: A blockchain containing blocks (Source: Chen et al., 2023)

2.2.5.1 Functional Characteristics

Ali et al. (2023) highlight key functional characteristics of blockchain that contribute to its growing adoption. First, blockchain is immutable, providing a permanent and tamper-resistant record of transactions that fosters trust across participants. Second, it is decentralized, as the ledger is replicated across multiple nodes in the network, eliminating a single point of failure and enhancing resilience. Finally, blockchain offers transparency, with a complete transaction history that enables provenance tracking, allowing the origin and movement of assets to be traced throughout their lifecycle.

2.2.5.2 Blockchain and Chain of custody Integration

Blockchain can provide a secure chain of custody for both digital and physical assets through its functional characteristics that facilitate transactions through trust, consensus, security, and smart contracts (Sultan et al.,2018). The security features of blockchain are especially useful when blockchain is used to store data that must be audited as well as requires a high level of integrity such as digital evidence. Moreover, access to data is recorded as a transaction thereby providing a reference to every activity that takes place in relation to the digital evidence stored. Finally, the blockchain property of interlinking blocks provides a credible source of audit trail which is useful in documenting the chain of custody of evidence gathered during digital forensics.

2.3 Empirical literature

2.3.1 Existing Forensic Chain of custody Solutions

2.3.1.1 Chain of Custody application using XML schema approach

Ratnasari et al. (2018) proposed a digital evidence chain of custody application utilising an XML-based approach. The application captures two types of chain of custody information: user-entered data and information extracted from the attributes of the digital evidence file. The XML method ensures the integrity of the evidence's hash value, making the evidence well-documented and acceptable in court. The MD5 value, representing the integrity of the digital evidence file, remains unchanged before and after the chain of custody information is entered (Ratnasari et al., 2018). In this approach, digital evidence and chain of custody files are stored in distinct repositories. The application automatically extracting digital file attribute information to link the evidence and its chain of custody records. The limitation of this approach is that digital evidence information is extracted from the digital file attribute value which is less accurate compared to metadata information of the digital evidence file.

2.3.1.2 Digital Evidence Cabinets (DEC)

To improve the management of digital evidence and the documentation of its chain of custody, Prayudi et al. (2014) introduced the idea of digital evidence cabinets (DEC). In

general, the idea of a digital evidence cabinet consists of the following: the tag cabinet concept, which is in charge of representing the digital evidence cabinet; the digital evidence management framework, which is in charge of managing the interactions between investigators at various stages of the investigation process; and the access control and secure communication concepts, which are in charge of offering assistance in terms of a reliable computing environment. While Digital Evidence Cabinets provide significant opportunities, they encounter the challenge of effectively managing large quantities of digital evidence. As the volume of stored data grows, performance problems arise compromising the overall efficiency of the chain of custody process.

Digital Evidence Cabinets compatibility issues further complicate the integration with existing forensic tools and digital evidence repositories (Bowman et al.,2019).

2.3.1.3 Smartcards

Shah et al. (2017) introduced the idea of employing smart cards to generate signatures and store forensic investigators' private keys to guarantee the integrity of digital evidence. An automated method extracts a comprehensive bit-by-bit image of the entire disk containing evidence and concurrently establishes a digital chain of custody, which is appended to the extracted image. A chain of custody is created in the form of rings with the initial chain of custody containing crucial details about the digital evidence. Rings get added to the chain of custody with each transfer of evidence during the forensic investigation process. The Smartcards approach has a limitation in that it can only store images in RAW format and not in multiple formats.

2.3.1.4 Digital Evidence Management Framework (DEMF)

Cosic et al. (2017) developed an application prototype based on the concept of digital evidence management framework (DEMF) presented earlier by Cosic and Baca in 2010. The DEMF solution allows recording and managing the chain of evidence at all stages of the digital forensic investigation while ensuring the integrity of the digital evidence.

It also enables packing of all the digital custody data together with digital evidence and ensures secure protection with the help of powerful AES256 encryption. Incase institutions or agencies need to exchange digital evidence, a.demf file containing all the

metadata of the complete digital life cycle is provided along with the evidence ultimately proving that the evidence has not changed or that its integrity has not been violated during digital forensic investigations. DEMF faces a challenge with scalability when it comes to handling extensive amounts of digital evidence across numerous cases or jurisdictions. Additionally, interoperability poses another limitation, particularly when integrating DEMF systems with pre-existing forensic tools, databases, and case management systems (Geradts & Verheij 2017).

2.3.1.5 A Blockchain-based Forensic Model for Financial Crime Investigation

Zarpala and Casino (2021) developed a blockchain-based forensic investigation framework designed to enhance the integrity and transparency of digital evidence management in financial crime cases, particularly embezzlement. The solution records and manages the chain of custody using smart contracts. Similar to Digital Evidence Management Framework (DEMF) systems, it packages forensic metadata with digital evidence to ensure provenance and non-repudiation. However, compared to DEMF, which relies on centralized encryption and secure evidence packaging such as .demf files, the blockchain-based model provides a decentralized alternative where smart contracts automate and verify each step in the forensic process with enhanced transparency and immutability. Nevertheless, like DEMF, it faces scalability issues due to the high volume of evidence, as well as interoperability limitations when integrating with existing forensic tools and legacy systems. A key limitation remains its narrow focus on financial crime, restricting its applicability to broader digital forensic domains (Zarpala & Casino, 2021).

2.3.1.6 Digital Chain of Custody Operational Framework

Pestana et al. (2023) proposed a standardized operational framework for the digital chain of custody (dCoC) aimed at enhancing the integrity, traceability, and legal reliability of digital evidence. The framework introduces two core concepts: Digital Custody Metadata (DCM), which captures the intent and history of evidence transfers; and Custody Transfer Points (CTPs), which mark the exact moments when custody of evidence changes. This approach addresses key challenges in digital evidence handling, such as data mutability, jurisdictional inconsistencies, and the lack of transparent documentation. Unlike

conventional methods that inadequately audit metadata or rely solely on file attributes, the proposed framework structure emphasizes custodianship accountability and standardized metadata documentation. The framework strengthens the admissibility of digital evidence by providing a legally auditable chain that is consistent across all stakeholders. Key challenges include harmonizing the framework across diverse legal jurisdictions and integrating it with existing digital forensic infrastructures (Pestana et al., 2023).

2.3.1.7 ForensicTransMonitor:

Alqahtany and Syed (2024) proposed ForensicTransMonitor a blockchain based framework designed to enhance the integrity, transparency, and traceability of digital forensic processes. The solution integrates all key phases of the digital forensic investigation process into a permissioned blockchain, with each forensic action immutably recorded to support a verifiable chain of custody. While the tool enhances forensic reliability, it is not specifically developed as a dedicated chain of custody management tools and therefore lacks specialised features for evidentiary handling, such as detailed custody tracking and legal validation. Moreover, it does not incorporate role-based access controls or identity verification mechanisms, which are essential for ensuring accountability in a chain of custody processes.

2.3.2 Summary of Gaps

Current solutions for managing the chain of custody of digital evidence have some drawbacks. These challenges include scalability issues when dealing with large amounts of evidence, difficulties in exchanging and collaborating on information due to incompatible data formats, and concerns about integrity since many existing solutions rely on manual processes or centralized databases, which are susceptible to human error, data entry mistakes, or tampering.

Table 2.2 below presents a summary of gaps of existing solutions

Table 2.2: Summary of gaps for chain of custody solutions

Authors	Focus of the Research	Research Methodology	Study Findings	Summary of Gaps
Ratnasari et al. (2018)	XML Approach for the Solution of Chain of Custody of Digital Evidence.	Prototype design, development and testing. /Development of an application prototype using XML schema, testing, and evaluation	Demonstrated that XML schema can effectively structure and manage chain of custody data	Digital evidence information is only extracted from the digital file attribute value which is less accurate compared to metadata information of the digital evidence file. Vulnerable to manipulation if the underlying database is compromised. Focus is on data structure, not security of the system itself.
Prayudi et al. (2014)	Digital Evidence Cabinets: A Proposed Framework for Handling Digital Chain of Custody.	Conceptual model and framework design.	Demonstrated that Digital Evidence Cabinets (DEC) can improve the organization and integrity of digital evidence	Limited focus on the chain of custody beyond storage. Integrity of evidence during transfer not addressed. Centralized storage creates a single point of failure. Limited scalability to effectively manage large quantities of digital evidence.
Shah et al. (2017)	Shah, S., et al. (2017). Digital Evidence Management System Using Smartcard Technology.	Design and implementation of a smart card-based system, testing, and evaluation.	Smartcards provide a secure platform for signing and verifying digital evidence.	Limited scalability to effectively manage large quantities of digital evidence. Challenges in integrating with existing forensic tools and digital evidence repositories
Cosic et al. (2017)	Digital Evidence Management Framework: Enhancing Chain of Custody in Digital Forensics.	Ontology-based framework development.	DEMF enhances chain of custody by standardizing metadata exchange and ensures evidence integrity.	Lacks practical implementation and assessment. Limited scalability. Interoperability limitation particularly when integrating with pre-existing forensic tools.

Zarpala & Casino (2021)	A blockchain-based forensic model for managing the chain of custody in financial crime investigations, especially embezzlement.	Conceptual framework and design of a decentralized forensic investigation system.	Demonstrated advantages over traditional DEMF systems through decentralization.	Narrowly focused on financial crimes, limiting broader application. Scalability issues due to high evidence volume. Lacks interoperability with existing tools.
Pestana et al. (2023)	Development of a standardized Digital Chain of Custody (dCoC) framework to improve the integrity, auditability, and legal admissibility of digital evidence.	Design of a conceptual operational framework	Demonstrated improvements in transparency, accountability and consistency.	Insufficiently handling of digital metadata. Limited mechanisms for custody accountability. Integration challenges with existing solutions.
Alqahtany & Syed (2024)	ForensicTransMonitor: A blockchain-based approach integrating the entire forensic investigation process.	Development of a permissioned blockchain framework for forensic data recording, with analysis of design and process.	Demonstrated improved transparency and process reliability across multiple forensic stages.	Not designed as a specialised chain of custody solution. Lacks role-based access control and identity verification. Insufficient legal validation mechanisms for detailed custody tracking for formal evidentiary use.

2.3.3 Conceptual Framework

The conceptual framework for the proposed chain of custody tool involves leveraging blockchain technology and smart contracts to establish a transparent, immutable, and auditable trail. Smart contracts enforce access control, evidence transfers, and other actions to ensure a secure and traceable log of all evidence-related activities. The framework defines three key user roles: Administrator, investigator, and legal users. Administrators manage user access permissions through a permission manager with permissions enforced by smart contracts to ensure that only authorised users interact with evidence. Investigators upload or transfer digital evidence by triggering a smart contract that immutably records the evidence details on the blockchain. Legal users request access to evidence with each request validated against predefined permission policies. All interactions with the evidence are transparently recorded on the blockchain, creating a verifiable audit trail. The resulting chain of custody timeline provides a chronological record of all evidence activities ensuring data integrity throughout the digital forensic process. Figure 2.3 below depicts the conceptual framework.

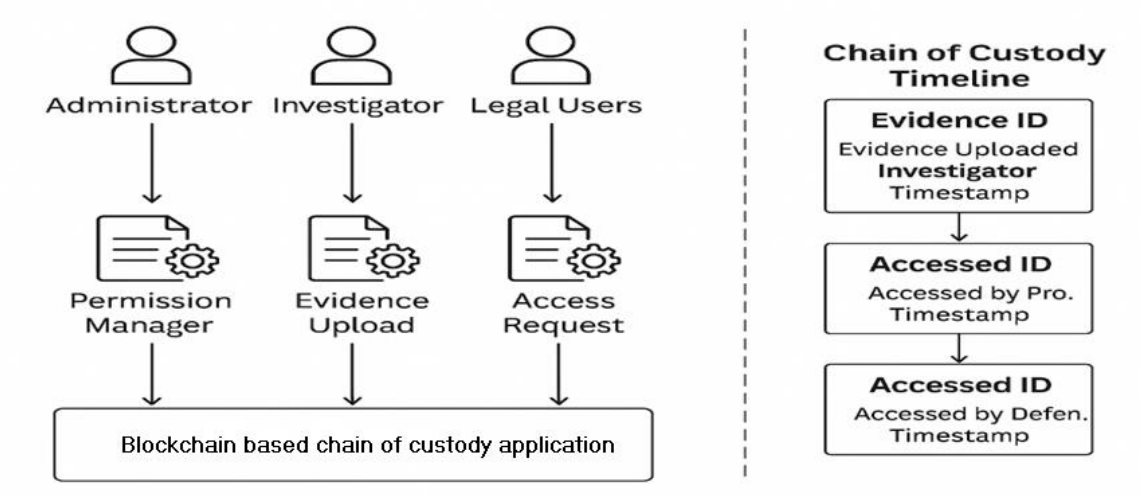


Figure 2.2: Proposed Conceptual Framework

Chapter 3: Research Methodology

3.1 Overview

This chapter outlines the methodologies employed to achieve the research objectives. The chosen approaches include Qualitative Research, Comparative Analysis, Design Science Research, and Agile Software Development.

3.2 Methodologies for study objectives

3.2.1 Qualitative Research

The first objective was to identify the common challenges that affect the integrity of digital evidence along the chain of custody. Qualitative research was selected for this objective due to its ability to explore complex issues, understand patterns, and gain insights. A comprehensive literature review concentrating on research papers related to digital forensics, digital evidence, and chain of custody challenges was conducted. This review highlighted the unique characteristics of digital evidence, such as ease of access, duplication, modification, and transfer, which complicate its documentation and preservation. Thematic analysis as guided by Braun and Clarke (2006) was performed to identify common themes and recurring challenges. The challenges were categorized into four main groups: technical, legal, personnel-related, and operational, as identified by Karie and Venter (2015). This approach provided a detailed understanding of the common challenges.

3.2.2 Comparative Analysis

The second objective was to review the existing solutions for preserving the integrity of evidence along the chain of custody. A comparative analysis was conducted to evaluate the effectiveness, advantages, and limitations of the solution focusing on key aspects such as scalability, security, interoperability, and ease of integration with existing forensic tools. A literature review of research papers, industry reports, and case studies was conducted to examine technologies such as blockchain, digital signatures, encryption, and secure logging mechanisms.

A summary of gaps of existing solutions was provided highlighting issues such as scalability, security and integrity due to reliance on manual processes or centralized databases. This review provided an understanding of the current state of solutions for preserving the integrity of digital evidence.

3.2.3 Design Science Research

Design science research is a methodology focused on creating innovative artifacts, such as systems or tools, to solve practical and real-world problems (vom Brocke et al., 2020). This methodology was used to achieve objectives three and four which focus on the design, development, testing, and validation of a blockchain-based solution. This methodology was selected because its key activities namely problem explication, requirements definition, design and development, and evaluation provide a structured framework for software development. This approach was crucial in identifying challenges and proposing a possible solution. The design and development phase of DSR also aligns closely with the iterative, flexible nature of Agile software development. The integration of DSR with Agile methodology enabled the development of a solution that met the objectives of the research.

3.3 Software Development Methodology

Building on the design science research methodology, software development was used to design and develop the system. A software development methodology "is a collection of procedures, techniques, tools and documentation aids which help the systems developers in their efforts to implement a new information system" (Avison & Fitzgerald, 2006). According to Geambasu et al. (2011), the success rate of software development projects is improved when a methodology suited to the project's specific characteristics is used. In this research, the Agile software development methodology was adopted within the DSR framework to guide the system's design, development, and testing phases.

Agile is a software engineering framework that starts with an initial planning phase and progresses towards deployment through iterative and incremental interactions throughout the project's lifecycle (Al-Saqqa et al., 2020).

The main goal of agile methodology is to reduce the extra effort and potential risks involved in software development by being flexible and adaptable to changes without compromising the overall process.

Agile methodology was preferred due to its incremental and adaptive nature in response to evolving requirements of block chain projects and flexible approach to development.
The phases of the agile methodology are as follows:

i. **System requirements phase:** During this phase, the objectives were defined, together with the system's functional and quality requirements. Users, developers, and system designers were all involved. The requirements covered the functional demands of the end user as well as the technical and physical attributes that define the engineering and operational boundaries.
ii. **System Planning Phase**: During this phase, a plan was created based on the provided requirements to determine the project's scope and define the application requirements
iii. **System design phase:** This phase encompassed the actual system development, including coding, testing, and integration activities. The requirements established in the initial phase served as a baseline for defining the system and subsystem specifications, outlining the system components, their interfaces, and the implementation process using the chosen hardware, software, and network resources.
iv. **System development phase**: In this phase, the system was built based on the requirements, design, and plan established in the previous phases. The developed system underwent thorough verification and validation through multiple testing cycles, including user acceptance testing, system testing, and unit testing.
v. **System release phase:** This phase marked the transition of the newly developed system into full operation and included the last round of user acceptance testing and the official sign-off by users.
vi. Additionally, thorough checks were conducted to ensure the system effectively met all the objectives and requirements outlined in the initial planning phase.

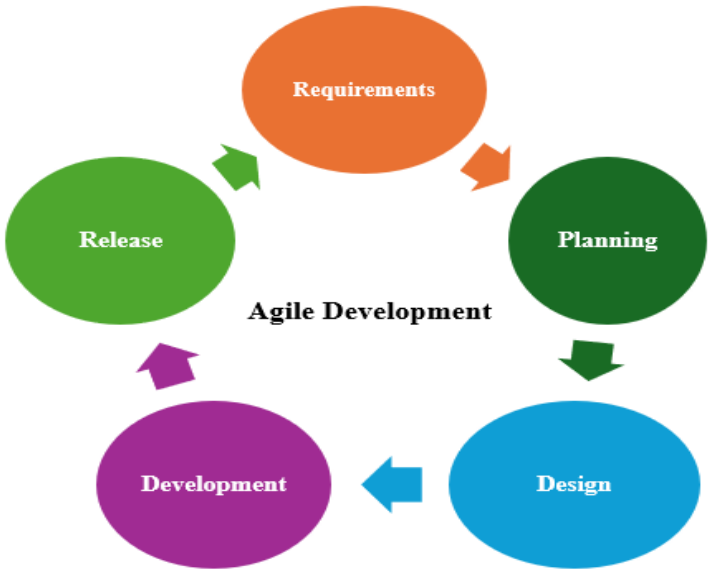


Figure 3.1: Agile Methodology (Sommerville,2015)

3.3.1 Requirements Phase

This phase focused on determining the various functional and non-functional requirements of the solution. The requirements were identified based on existing technologies for maintaining a digital chain of custody. The requirement analysis for this study was based on a thorough examination of the gaps identified in current chain of custody solutions. A comprehensive review of existing literature on chain of custody management and the adoption of blockchain technology in digital forensics was conducted to formulate a solution that enhances chain of custody integrity. This phase defined and documented the technical approach used to successfully develop the proposed solution.

3.3.2 System Planning Phase

This was achieved by examining the existing literature to gain an understanding of how existing solutions work the functionality and areas covered, as well as the aspects not addressed. The information collected was used to identify the system requirements, system prerequisites, project scope and limitations.

The requirements were assessed to ensure they are feasible. Only qualifying features were included in the system, while extras were excluded.

3.3.3 Analysis and Design

Analysis and design modelling in system development is important to ensure the quality of both the process and product. This study employed an object-oriented analysis and design (OOAD) methodology. Object-oriented analysis examines a problem domain defined as a set of use cases to extract classes that define the problem (Pressman, 2005).

In Object-Oriented Analysis and Design (OOAD), the analysis phase utilises Unified Modelling Language (UML) to create a representation of the real-world application, emphasizing its essential characteristics. The Unified Modelling Language (UML) is a general-purpose visual modelling language used to specify, visualize, construct, and document the artifacts of a software system (Object Management Group [OMG], 2017). It also serves to capture decisions and enhance understanding about systems that must be constructed and used to design information systems (Jacobson et al., 2021). Below is a detailed description of the UML diagrams that were used:

3.3.3.1 Use Case Diagram

A use case is a coherent unit of functionality expressed as a transaction among actors and the system (Jacobson et al., 2021). The purpose of the use case view is to list the actors and demonstrate the actors' involvement in each use case. Use case diagrams were used to partition to partition the proposed solution into distinct functionalities and aid in identifying the system's functional requirements.

3.3.3.2 Sequence Diagram

A sequence diagram shows a set of messages arranged in time sequence. Sequence diagrams show how events or activities in a use case are mapped into operations of object classes in the class diagram (Al-Fedaghi,2021). Sequence diagrams model the interactions and collaborations between objects in the order they occur illustrating the application's logical flow and enabling the documentation and validation of its logic. In this study, sequence diagrams were used to refine the requirements expressed in the use cases and visualize the interactions between various components of the system.

3.3.3.3 Design Class Diagram

Design class diagrams were used to model the application, representing the classes within the system along with their attributes, operations, and relationships. A class is an abstraction that defines the common structure and behaviour shared by a group of objects. Objects, which are instances of these classes, are created, modified, and destroyed during the system's execution. These diagrams offer an in-depth overview of the system's structure, including the classes, their attributes, methods, and the relationships among them.

3.4 Implementation

Ethereum was selected for the study due to its robust smart contract capabilities, strong developer support, and its ability to support a transparent and auditable chain of custody solution. It was also chosen for its compatibility with test networks such as the Sepolia Testnet, which provided a cost-free environment for smart contract deployment and execution, making it ideal for iterative development during the proof-of-concept phase.

The solution was developed using Etherscan. Etherscan is a blockchain explorer and analytics application specifically designed for the Ethereum blockchain and widely used to obtain insight into blockchain activity, track transactions and smart contracts. Etherscan provides an API that developers can use to build applications and services on top of Ethereum blockchain data and a variety of tools and features to explore and search the Ethereum blockchain for transactions, addresses, tokens, smart contracts, and other network events. Etherscan was used for development, packing, deployment, and testing of the system

3.5 Testing

Software testing is the process of evaluating system components against specified requirements, either manually or using automation tools, to identify discrepancies between expected and actual results (Hooda & Chhillar, 2015). Testing was conducted to ensure that the tool fully meets both the functional and non-functional requirements. Agile testing methodology was employed as it enabled continuous testing from beginning of

development to deployment. The following tests were undertaken to ensure system functions as intended.

i. Usability Testing

Usability testing is an evaluation method in which one or more representative users at a time perform tasks or describe their intentions under observation (Riihiaho,2018). The main purpose of the usability testing in this study was to confirm that the proposed solution is suitable for real-world investigation scenarios. Conducting usability testing provided a dependable method for quantitatively evaluating users' performance and their subjective satisfaction with the system.

ii. Functionality Testing

Functional testing involves verifying that a software system operates as intended under specified conditions, ensuring compliance with user requirements (Ammann & Offutt, 2016). The functional testing focused on key areas including system access control, evidence registration, evidence storage and security, transaction logging and timestamp verification, and immutability

iii. Compatibility Testing

During the compatibility testing, various tests were conducted, with the browser compatibility test being of utmost importance. This test focused on assessing the compatibility of two major browsers, namely Firefox (version 124.0.1 (64-bit)) and Microsoft Edge (Version 123.0.2420.65 (Official build) (64-bit)). The purpose was to determine whether the system was compatible with the selected browsers.

iv. Integration Testing

Integration testing is a systematic technique for assembling a software system while conducting tests to uncover errors associated with interfacing (Naik & Tripathy, 2011)

Integration testing is a systematic method of assembling a software system while executing tests to identify interfacing issues (Naik & Tripathy, 2011). Integration testing ensures that different components (modules, classes, or services) work together

seamlessly within the larger system. The main objective of integration testing in this research was to validate the functioning of the system and detect any errors during the interaction between combined components.

3.6 Ethical Considerations

This research appropriately recognised and cited all external concepts within the text to acknowledge the data sources and their contributions. No real-world data was used in this study, minimising potential data privacy risks.

Chapter 4: System Design and Architecture

4.1 Overview

This section outlines the architectural design of the blockchain-based chain of custody tool, highlighting its functional and non-functional requirements. The system's core functionality includes adding, displaying, transferring, and reporting evidence, with all actions recorded immutably on the blockchain. A web application allows authorized users to interact with a backend that securely transmits transactions to the blockchain. Blockchain ensures a tamper-proof log of system events, providing an auditable, chronological record of all actions. Unlike traditional forensic tools that rely on centralized databases, this system utilized decentralized consensus, cryptographic hashing, and real-time transparency to ensure data integrity. This enhanced trust in digital forensic investigations and offered a comprehensive chain of custody mechanism, ensuring the security and authenticity of evidence throughout its lifecycle.

4.2 Functional Requirements

A functional specification is a description of the expected behaviour of the program. Functional requirements outline how users interact with the system and what the system should accomplish to be considered efficient. The tool's functional requirements include:

1. Evidence Chain of Custody Documentation
    a. The system must allow authorized users to register digital evidence, capturing key metadata such as timestamp, source, type, and associated case details.
    b. Each interaction with the evidence (e.g., access, modification, transfer) must be immutably logged on the Ethereum blockchain to ensure traceability.
2. Security of the chain of custody
    a. The system must leverage cryptographic hashing to prevent tampering of digital evidence records.
    b. Any changes or transfers of evidence must be recorded as blockchain transactions, ensuring auditability and accountability.
3. Access control and user authentication.

a. Users must authenticate securely via login credentials, with all access attempts logged on the blockchain.
b. The system must implement access control based on user roles (such as investigators, defence) being used to place restrictions.

4. Availability of secure reports.
   a. The system must generate and store immutable reports detailing the complete history of evidence handling.
   b. Authorized users should be able to retrieve evidence reports that provide verifiable proof of authenticity for legal and forensic proceedings.

4.3 Non-Functional Requirements

The non-functional requirements outline the system's attributes that enable it to efficiently meet the functional requirements. They include the following areas:

1. User Interface and Experience
   a. The web application must provide an intuitive and user-friendly interface.
   b. The system should display evidence records clearly, enabling easy tracking of chain of custody events (e.g., transfers, modifications).
2. Performance and Accuracy
   a. The system must ensure high accuracy in recording and retrieving blockchain transactions to maintain uncompromised evidence integrity.
   b. Performance should be optimized to handle multiple concurrent users without degradation in response time.
3. Resource Efficiency and Scalability
   a. The system must efficiently utilize computational and storage resources, minimizing transaction costs on the Ethereum network.
   b. The architecture should support future scalability to accommodate growing evidence records and additional forensic functionalities.
4. Code Quality and Reliability: The system must handle blockchain transactions efficiently without data loss or integrity breaches.

4.4 System Architecture

This section provides a high-level design of how the blockchain-based chain of custody system operates. The illustration in Figure 4.1 depicts the high-level architecture design of the system.

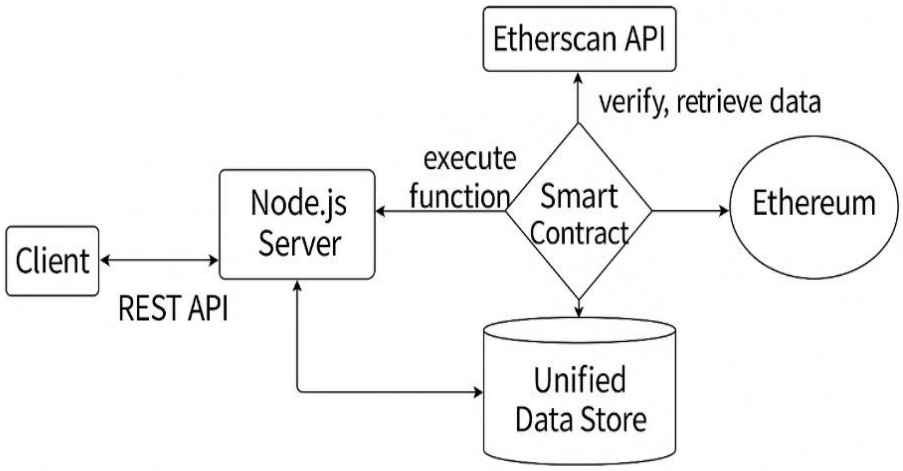


Figure 4.1: System Architecture

The main elements consist of a client that interacts with a central Server via an API. The Server acts as middleware, processing client requests and orchestrating interactions between Smart Contracts and a unified Data Store. Smart Contracts, connected to the Blockchain, execute on-chain logic and immutably record evidence hashes. The Etherscan Blockchain Explorer provides read-only access for verifying on-chain transactions and metadata.

4.5 Roles and Functionalities

Table 4.1 below highlights the key roles, their associated responsibilities, and functionalities within the system. Each role was designed to ensure the effective management and integrity of digital evidence throughout its lifecycle. The corresponding functionalities were implemented through specific operations that supported various tasks, ranging from user management to evidence handling and reporting.

Table 4.1: Roles Overview

Role	Responsibilities	Functionalities
Admin	Register users, manage cases and evidence, grant access, oversee system security	Create/manage user accounts, register cases, add evidence, grant access, generate reports
Investigator	Access and review evidence, manage evidence handling and transfer of evidence	Access case, Retrieve evidence details, transfer, access reports
Defence	Access and review evidence, ensure evidence authenticity for defence cases	Retrieve evidence details, access reports
Prosecutor	Access and review evidence for prosecution, ensure evidence authenticity	Retrieve evidence details, access reports

4.6 Inputs, Processes and Outputs

This section provides an overview of the system's inputs, processes, and outputs. It explains how various types of data and user interactions are handled, processed, and transformed into valuable outputs.

4.6.1 Inputs

The system receives various types of input data related to digital evidence and user interactions. These inputs include:

1. Case Registration Data:
    a. Case ID
    b. Timestamp of creation
    c. Add or remove Assignee(s)

2. Evidence Registration Data:
    a. Evidence ID
    b. Evidence Type (e.g., document, image, video, log file)
    c. Timestamp of collection
    d. Source of evidence (who collected it and from where)
    e. Case reference number
3. User Authentication and Access Requests:
    a. User login credentials
    b. Role-based access control
    c. Actions performed on evidence (e.g., view, transfer, modify)
4. Evidence Transfer Requests:
    a. Sender and recipient details
    b. Timestamp of transfer
5. Report Generation Requests:
    a. Query parameters for retrieving chain of custody records
    b. Filtering criteria (e.g., by date, case ID, evidence ID)

4.6.2 Processes and Algorithms

The system applied various processes to ensure the integrity, security, and auditability of digital evidence using blockchain technology. These processes include:

1. Evidence Registration and Hashing:
    a. When evidence is added, is assigned a unique address on the blockchain. This address acts as its fingerprint. Any modification would merely be recorded as new entries.
    b. Ethereum entries ensure data stored on the blockchain is immutable and authentic.
2. Blockchain Transactions for Chain of Custody Logging:
    a. Every action (evidence creation, modification, access, transfer) is recorded as a blockchain transaction.

   b. Smart contracts enforce the logging of transactions to maintain the chronological history of evidence handling.
3. Access Control Enforcement:
   a. Role-Based Access Control (RBAC) verifies user permissions before granting access to evidence.
   b. Unauthorized access attempts are logged for security monitoring.
4. Evidence Integrity Verification:
   a. Users can verify that evidence has not been altered by comparing its current hash with the original blockchain-stored hash.
   b. Any mismatch triggers an integrity alert.
5. Audit Trail and Report Generation:
   a. The system retrieves chain of custody records from the blockchain.
   b. Users can generate reports detailing the full history of evidence handling.

4.6.3 Outputs

The system produces various outputs based on user interactions and system processes:

1. Immutable Chain of Custody Records: A blockchain-stored log of all evidence interactions, ensuring auditability.
2. Verification Status of Evidence Integrity: A confirmation that evidence remains unaltered, or an alert if integrity has been compromised.
3. Access Logs and Security Reports: Detailed records of user activities, including evidence access and modification attempts.
4. Forensic Audit Reports: Chronological reports of evidence handling, exportable for legal and investigative purposes.

4.7 Prototype Structure

The proposed system serves as a prototype designed to demonstrate the feasibility of managing evidence chain of custody using blockchain technology. The system consists of four fundamental functions: Adding, Displaying, Transferring, and Reporting evidence information from the Blockchain.

In the illustration provided in Figure 4.2, a process for managing cases and evidence is demonstrated. The process begins with the user attempting to log in. The system verifies the success of the login attempt. If the login is unsuccessful, the process returns to the Login step. However, if the login is successful, the process proceeds to the next step.

The system then checks for the presence of existing cases. If there are no existing cases, the user is directed to Create Case. On the other hand, if there are existing cases, the user proceeds to Select Case. Once a case is selected, the system verifies if there is any existing evidence. If there is no existing evidence, the user is required to Add Evidence. Conversely, if there is existing evidence, the user can Select Evidence for viewing.

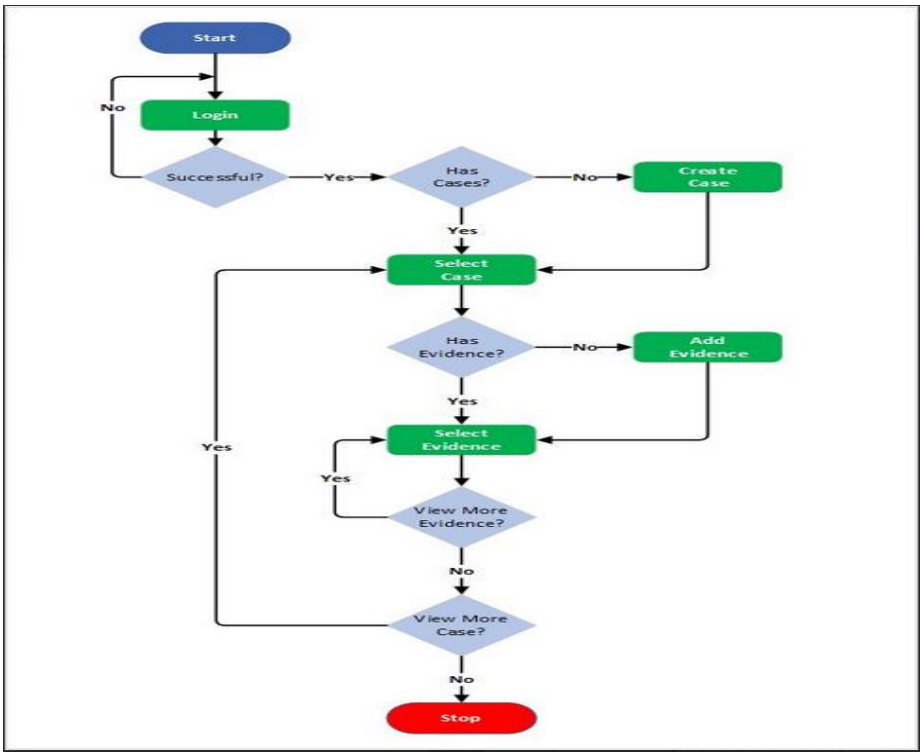


Figure 4.2: Prototype Structure

4.8 System Design Tools

This section presents the system design tools used to visually represent and model the architecture and functionality of the system

4.8.1 Use Case Diagram

A use case diagram is an example of a Unified Modelling Language (UML) diagram that presents the communication between actors (users or external systems) and a system. It offers an overview of the system's functionality from the end user's perspective by demonstrating external behaviour and interactions. In Figure 4.3 below, there are two actors identified. The admin can add and manage users, cases and evidence. Other users (Investigator, Defence, Prosecutors or other permissible user) interact with the system to access cases and evidence. The restrictions ensure security and controlled access to data. All users must authenticate before accessing any system functionality. All actions including login ends in the addition of a blockchain entry for immutability.

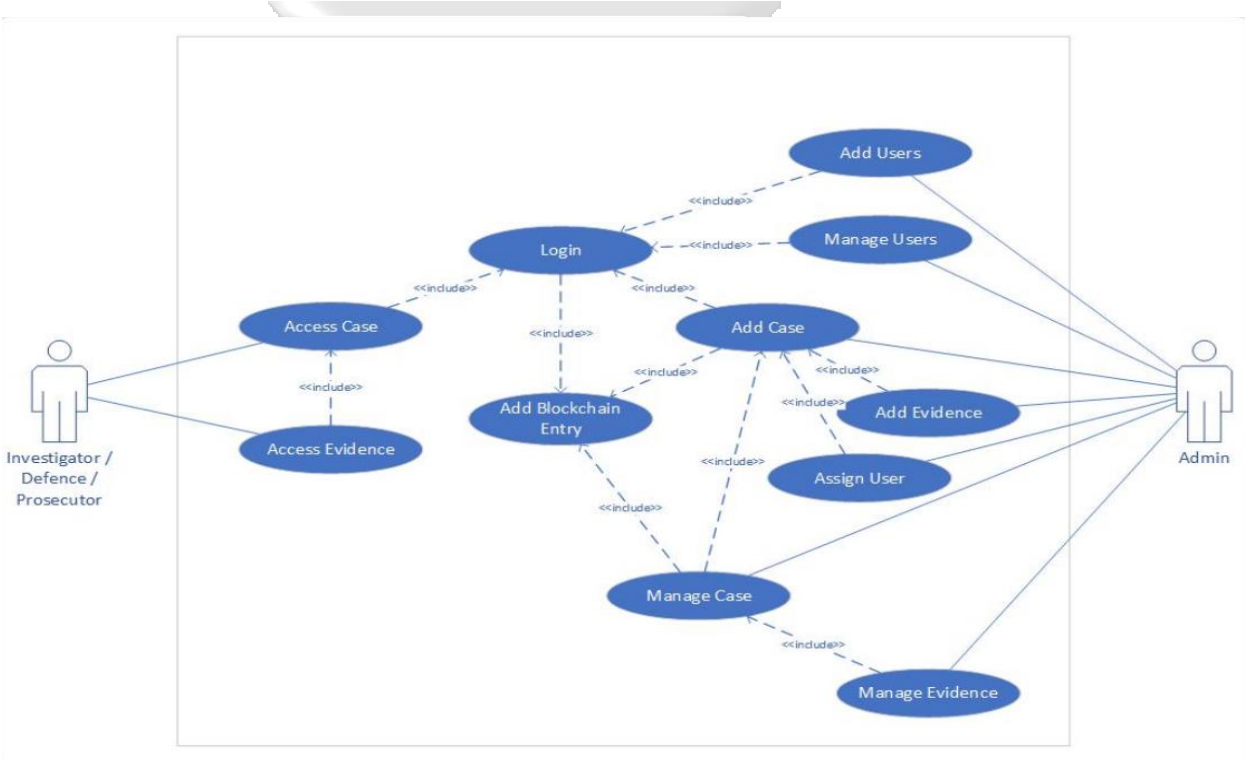


Figure 4.3: Use Case Diagram

4.8.2 Sequence Diagrams

4.8.2.1 Administrator

The sequence diagram in the Figure 4.4, illustrates the flow of interactions between the Administrator, Node Server and blockchain network within the system.

The sequence begins with the Administrator sending a “Login Request” to the Node Server. The Node Server records this login action on the Blockchain. Upon successful login recording, the Node Server responds to the Administrator with a “Login Success” message. Next, the Administrator can “Add Case,” which is again processed by the Node Server and recorded as an action on the Blockchain.

Similarly, when the Administrator adds evidence, it’s processed by the Node Server and recorded on the Blockchain. There are also steps for requesting case details and evidence details these steps involve interactions with the Node Server and actions recorded on the Blockchain.

Figure 4.4: Admin Sequence Diagram

4.8.2.2 User (Investigator /Prosecutor)

The sequence diagram in the Figure 4.5, illustrates the flow of interactions between the User, Node Server and blockchain network within the system. The sequence begins with the User sending a “Login Request” to the Node Server.

The Node Server records this login action on the Blockchain. Upon successful login recording, the Node Server responds to the User with a “Login Success” message. Next, the User requests case details from the Node Server. The Node Server in turn requests the case entry from the Blockchain. Once received, the Node Server sends back the “Get Case Details” response to the User. Additionally, when evidence details are requested by the User, this request is sent both to the Node Server and directly to the Blockchain. Both the Node Server and the Blockchain respond with their respective “Get Evidence Details” and “Get Case Entry” messages.

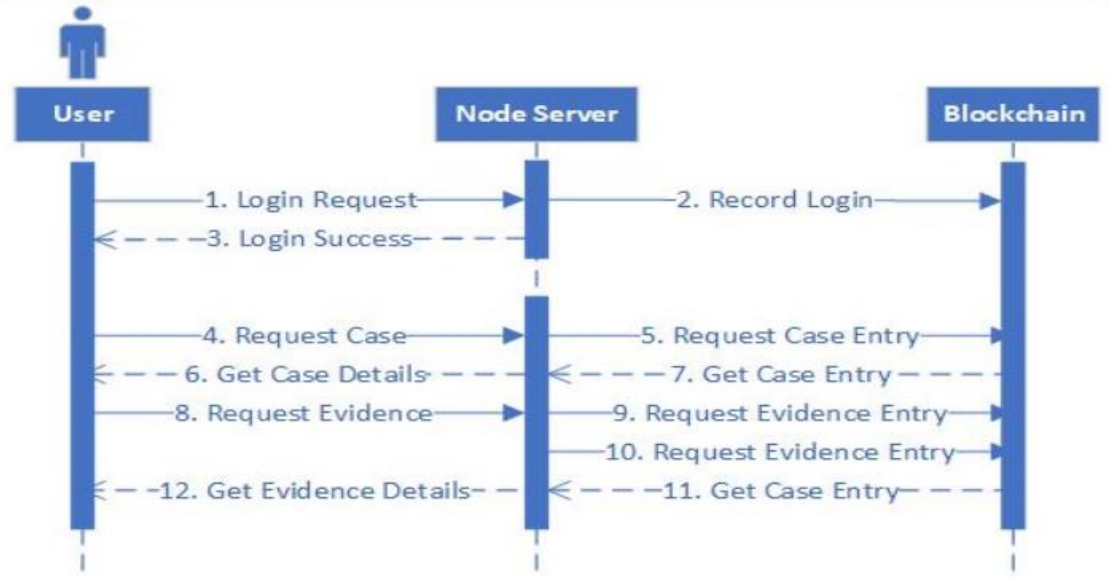


Figure 4.5: Admin Sequence Diagram

4.8.3 Class Diagram

The class diagram in the Figure 4.6 represents the interactions and dependencies among these entities in the system. The diagram main classes/entities associated attributes (properties) and relationships with other classes. The classes and their attributes are described as below:

1. Admin
   i. Attributes:
      a. address: Unique identifier for the admin.
      b. index: Position or identifier within the system.

c. password: Authentication credential.
d. isActive: Indicates whether the admin account is active.

ii. Responsibilities:

a. Manages Users and Cases.

2. Users

i. Attributes:

a. address: Unique identifier for the user.
b. index: Position or identifier within the system.
c. password: Authentication credential.
d. isActive: Indicates whether the user account is active.

ii. Responsibilities:

a. Views Cases and Evidence.

3. Case

i. Attributes:

a. id: Unique identifier for the case.
b. status: Represents the status of the case (e.g., open, closed, under investigation).

ii. Responsibilities:

c. Managed by both Admin and User.
d. Contains Evidence.
e. Viewed by Users.

4. Evidence

i. Attributes:

a. id: Unique identifier for the evidence.
b. name: Name or title of the evidence.
c. timeAdded: Timestamp of when the evidence was added.
d. timeUpdated: Timestamp of the last update.
e. details: Additional metadata or description.

ii. Responsibilities:

f. Each Case contains one or more pieces of evidence.

g. Viewed by Users.

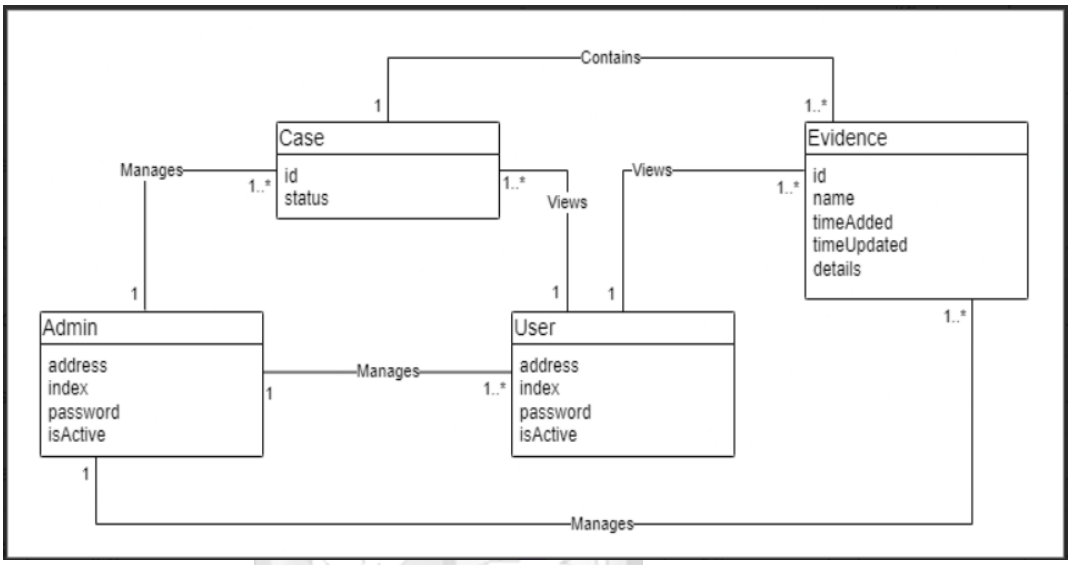


Figure 4.6: Class Diagram

4.8.4 Activity Diagram

The activity diagram in the Figure 4.7 represents the interactions and dependencies among these entities in the system. It shows case and evidence management process within the system. It outlines the logical sequence of actions taken by a user to log in, create or manage cases, and view or add evidence.

The process begins at *Start*. A user will the attempt to *Login* into the system. If *Successful*, the user will proceed to check if they have cases (*Has Cases?*). The user can then *Create Case* if none exist or *Select Case* to view. They will the look at whether there is evidence (*Has evidence?*) from where they can either *Add Evidence* or *Select Evidence* to view. They will loop back to *Select Evidence* if they want to *View More Evidence* or proceed to check if they will *View More Cases*. The decision here is to either loop back to *Select Case* or simply end the process in *Stop*.

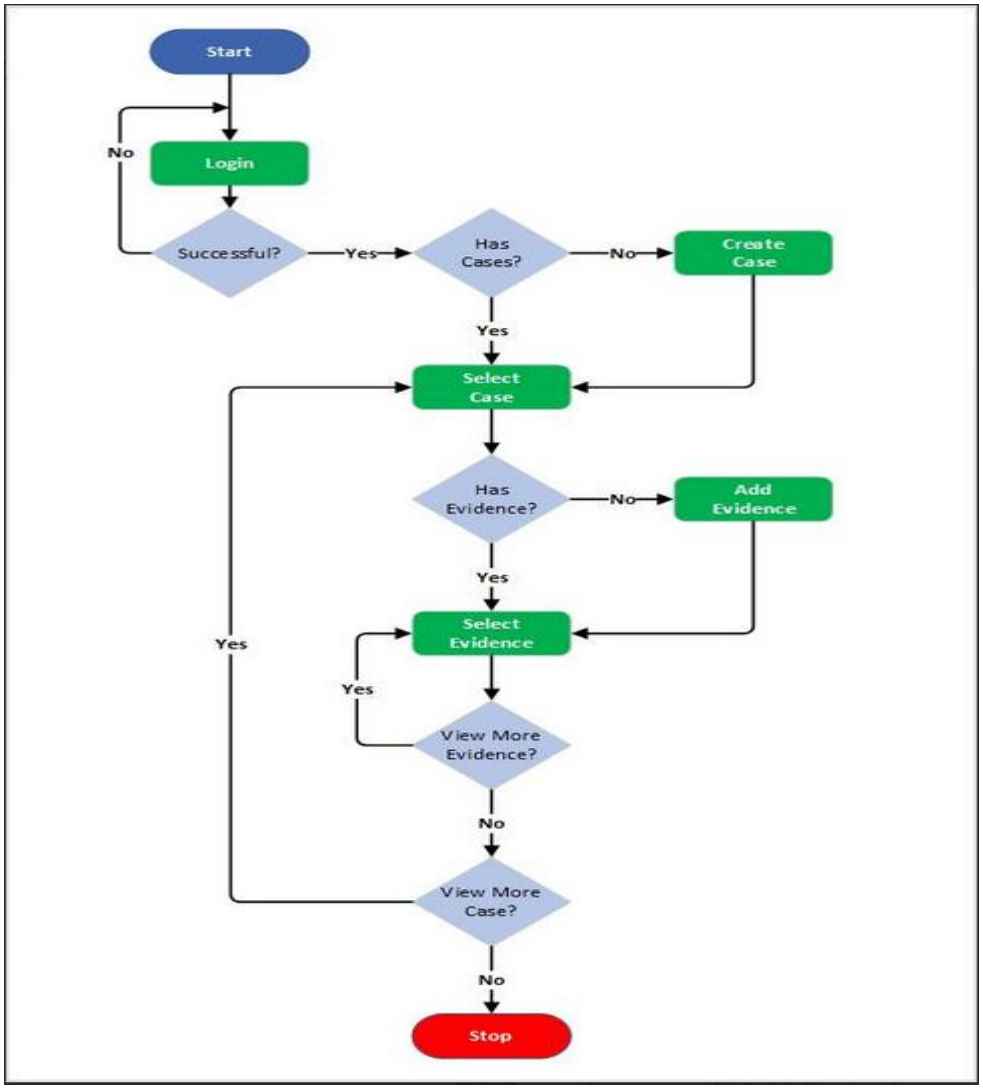


Figure 4.7: Activity Diagram

4.9 Security design

The security design of the system followed a multi-layered approach to ensure data confidentiality, integrity, and availability. This design incorporated security principles and architectural patterns aimed at mitigating risks and enhancing system resilience.

4.9.1 Security Protocols and Mechanisms

The system employed several security protocols and mechanisms to enforce protection at different levels:

1. Authentication and Access Control
    a. JWT (JSON Web Token): Secure token-based authentication ensured that only authorized users could access system resources.
    b. Role-Based Access Control (RBAC): Enforced the principle of least privilege by restricting access based on predefined user roles.
2. Data Security
    a. Secure Communication (HTTPS & End-to-End Encryption): All data transmitted between the user's browser and the server was protected using HTTPS. This ensured that any information sent, such as login details, case files, or evidence, could not be intercepted or altered by unauthorized parties.
    b. Blockchain Immutability: Transactions and evidence records were permanently stored on the blockchain, preventing tampering.
    c. Data Hashing: Cryptographic hashing was utilized to ensure data integrity, allowing verification of stored information. Blockchain helped achieve this as any transaction with immutable data was also attached with a cryptographic hash that could be used to audit the data.
3. Smart Contract Security: Access Control in Smart Contracts ensured only privileged users (e.g., administrators) could perform specific blockchain operations.

4. Secure Communication: HTTPS (SSL/TLS 1.3): Ensured encrypted communication between the web application and backend APIs.

5. Logging and Monitoring

    a. Blockchain Logs: Every transaction was immutably logged on the blockchain, creating a tamper-proof audit trail.

    b. Security Event Logging: The system logged user activities and access attempts, aiding in intrusion detection.

4.9.2 Security Principles

The following security principles were integrated into the system architecture:

1. Defence in Depth: Security controls were implemented at multiple layers, including authentication, authorization, and data validation.
2. Least Privilege Access: Users were granted only the necessary permissions based on their roles to minimize exposure to sensitive data.
3. Separation of Concerns: Security mechanisms such as authentication, authorization, and data protection were handled separately.
4. Auditability: Every transaction and modification within the system was immutably logged on the blockchain, ensuring accountability and traceability.

4.10 Application Wireframes

This section includes wireframes to illustrate the design of the system and key features. The purpose is to provide a clear conceptual view of the system and ensure alignment with functional requirements

Login Page

Users can log into the system by entering their username and password on the login page, as illustrated in Figure 4.8

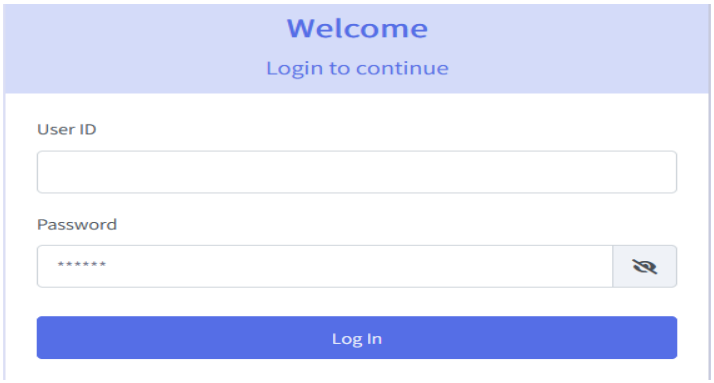


Figure 4.8: Login Page Wireframe

Add Case

Every case needs to be registered requires to be registered with a case ID for tracking. Figure 4. 9 below shows registration for a new case.

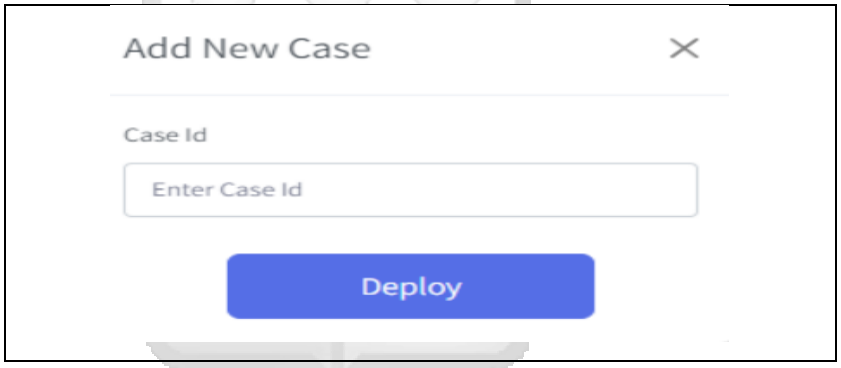


Figure 4.9: Add Case Wireframe

Add Evidence & Details

For each case, relevant evidence is added with evidence ID for tracking & evidence details. Figure 4. 10 below shows evidence addition.

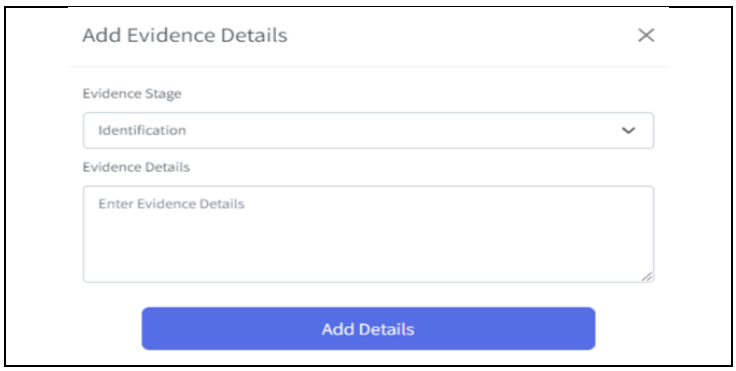


Figure 4.10: Add Evidence & Evidence Details

Chapter 5: System Implementation and Testing

5.1 Introduction

This chapter addresses the implementation of the chain of custody tool and emphasizes its key functions. This section contains screenshots of notable features and tests conducted on the system. The system was implemented using the tools described in Chapter Three of this study.

5.2 System Requirements

5.2.1 Hardware Requirements

The system required a combination of client-side and server-side hardware to support case and evidence management efficiently.

1. Client-Side (User Devices)
    a. Desktop or laptop (Windows, macOS, Linux)
    b. Minimum 8 GB RAM, Intel i5 (or equivalent) processor
    c. Stable internet connectivity
2. Server-Side (for hosting the system)
    a. Cloud-based or on-premises server
    b. 2GB+ RAM, Duo-core processor or higher
    c. Security measures such as firewalls and access controls

5.2.2 Software Requirements

The system comprised of multiple software components that facilitate its functionality. These include:

1. Operating System:
    a. Ubuntu 20.04+ (for hosting the API server and blockchain nodes)
    b. Windows 10+ (for development and testing)
2. Backend Development:
    a. Node.js – Provides the runtime environment for the API server
    b. Express.js – Framework for handling API requests and responses
    c. Ethers.js – Library for interacting with the Ethereum blockchain

3. Frontend Development:
    a. React.js – Used to build the web application interface
    b. Axios – Facilitates API communication with the backend
4. Blockchain & Smart Contracts:
    a. Solidity – Programming language for writing smart contracts
    b. Ethereum (Sepolia Testnet) – Blockchain network used for deploying contracts
    c. Etherscan API – Used for contract verification and blockchain data retrieval
5. Authentication & Security:
    a. JWT (JSON Web Token) – Secures API authentication
6. Development & Deployment Tools:
    a. Postman – For API testing and debugging
7. Version Control & CI/CD:
    a. Git & GitHub – Version control for collaborative development

5.2.3 Network Requirements

Reliable network infrastructure ensures secure data transmission and system accessibility. Below are the minimum requirements:

1. Internet Connection: Minimum 10 Mbps for smooth operations
2. Hosting Environment: Cloud-based (AWS, Azure, Google Cloud) or premise. deployment. Google Cloud was chosen for this implementation.
3. Encryption: HTTPS/TLS for secure communication
4. API Communication: RESTful API to facilitate interactions between frontend, backend, and blockchain components
5. Firewall & Security: To prevent unauthorized access and data breaches.

5.3 System Components

5.3.1 Application Programming Interface Server

Application Programming Interface or (API), is a set of rules, protocols, and tools that allow different software applications or components to communicate with each other. It defines the methods and data formats used to interact with a particular software component or service.

In this system, the API server played a pivotal role in connecting different system components, managing data store and blockchain interactions, and facilitating communication between the Web Application and other system parts.

5.3.2 Smart contract

Smart contracts are scripts that are stored on the blockchain and become active when a transaction is directed towards them. These contracts operate independently and automatically across all nodes within the network, executing a predetermined set of instructions based on the information provided in the triggering transaction. Smart contracts are essential components in the system, as they establish the rules of the underlying Ethereum blockchain infrastructure.

The system employed various contracts, such as the admin contract responsible for managing user account access to the blockchain, the Case factory contract which handled the deployment of cases and evidence, and the Case Contract which facilitates case access and logs storage. The deployment of smart contracts involves three components: ETHERSCAN_API_KEY, which is utilized for smart contract verification, MNEMONIC a 12–24-word phrase for generating private and public key pairs for blockchain access, and SEPOLIA, a node through which the blockchain network is accessed. The Etherscan platform is used to access the Ethereum Blockchain.

5.3.3 Web Application

This component served as the entry point for human interaction. The web application was built using React JS and was made available for access through a web browser. Users were provided with a graphical interface to interact with, which abstracted the other layers of the system from them. The web application interacted with the server through APIs.

5.4 System Modules

5.4.1 Registration and Authentication

Users access the system through authentication. They however need to have been registered first before they can authenticate. Only the system administrator can register new users to ensure that only authorised users have access to the tool. To register a user, a request with user data is posed to the server after which the server uses a smart contract to create a user entry on the blockchain. This functionality can be seen in Appendix A.

Once Registration has been done, a user can then login from web application. The login request sent from the browser is recorded therefore providing a log entry that can be checked later. Figure 5.1 below shows the transactions recorded at a specific time.

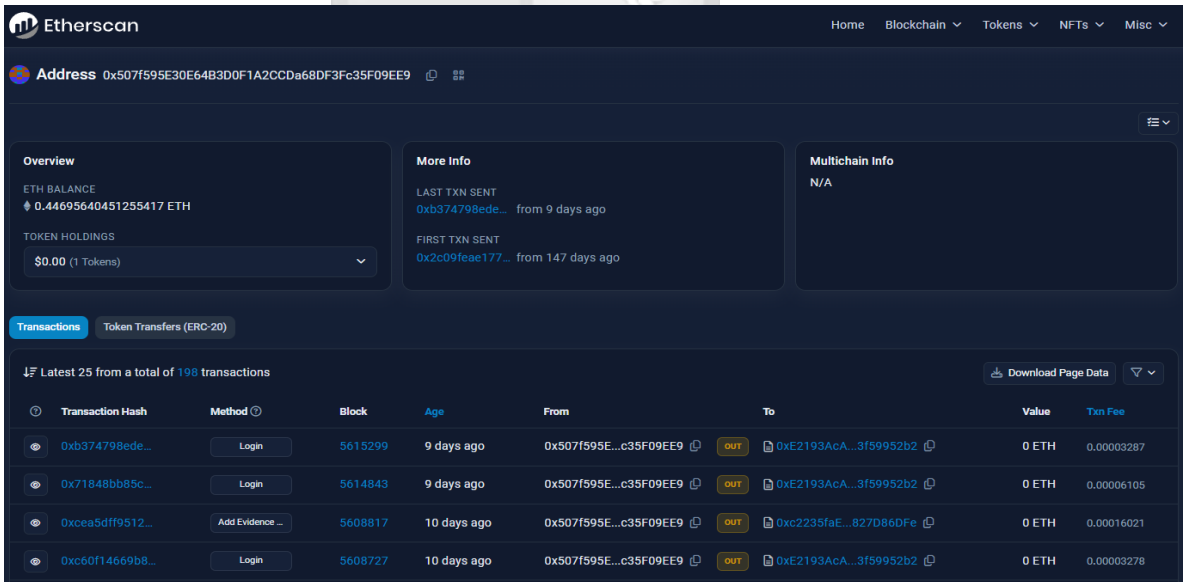


Figure 5.1: Registration Blockchain Transaction

A further look into the login record shows several details that provide more insight about the event being recorded as shown in Figure 5.2

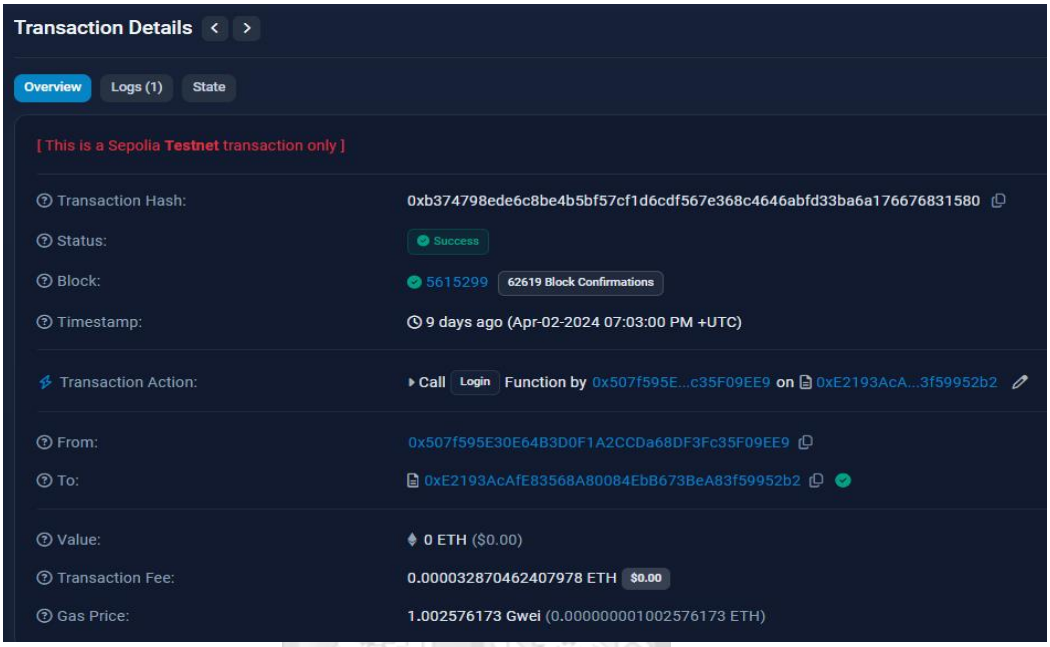


Figure 5.2: Blockchain Login Transaction

5.4.2 Add Case -Administrator

On the web application, a user is provided with a simple form to add a case ID. A blockchain entry about the creation of a case follows up. Figure 5.3 and Figure 5.4 below both the user interface and blockchain verification of recorded events.

Figure 5.3: Add Case Form

Figure 5.4: Blockchain Log for Added Case

5.4.3 Add Evidence -Administrator

After creation of a case, evidence pertaining to a case is added. Once evidence has been setup, its details can now be added according to the different stages of a case. The subdivision will help keep track of the evidence changes as the case proceeds therefore creating a chain of custody record. Figures 5.5,5.6 and 5.7 below show the evidence form for adding evidence, blockchain transaction and blockchain log.

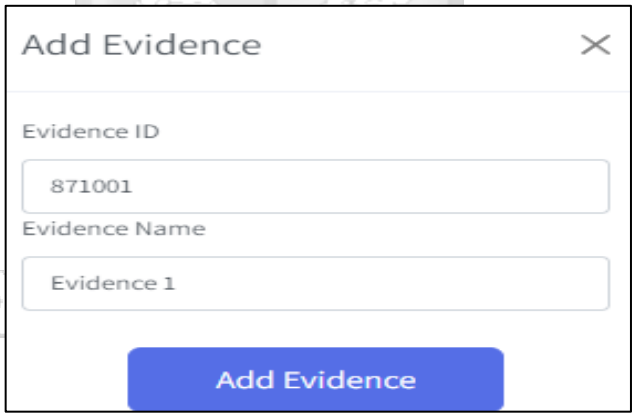


Figure 5.5: Add Evidence Form

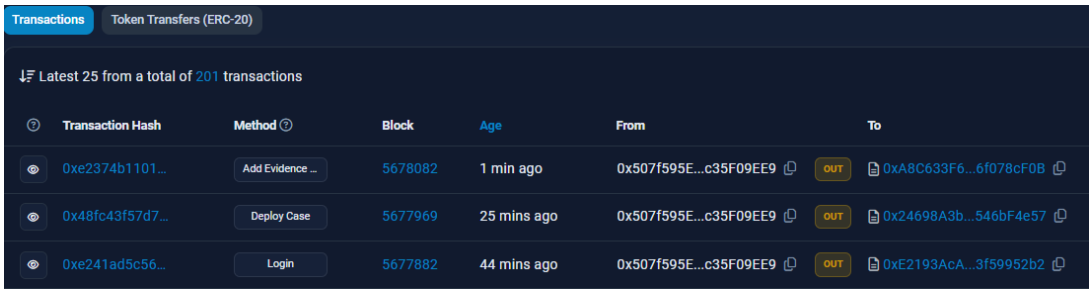


Figure 5.6: Add Evidence Transaction Log

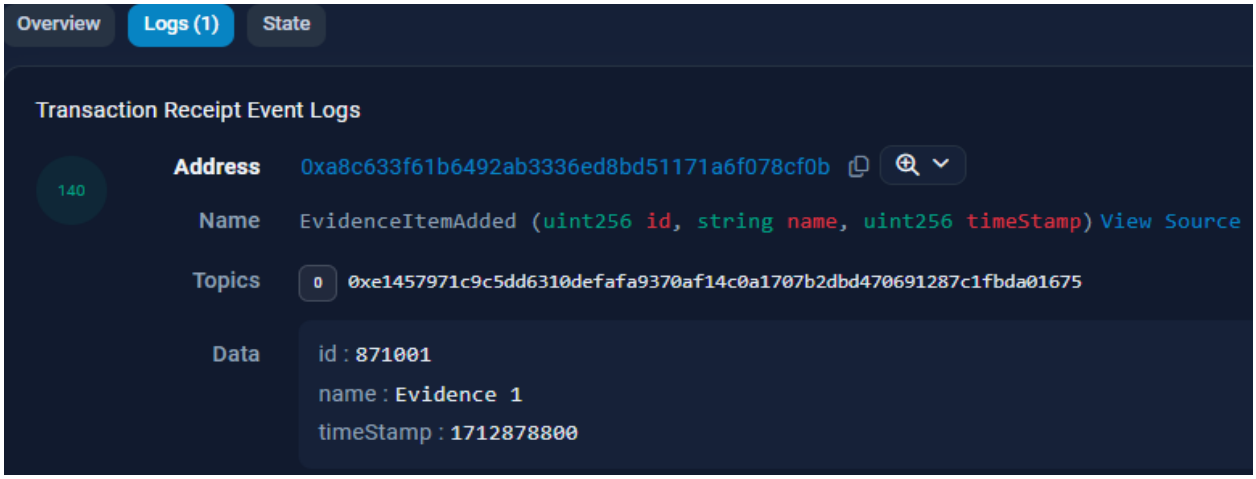


Figure 5.7: Add Evidence Blockchain Log

5.4.4 Assign Access to a Case -Administrator

System administrators can add cases and evidence. Any other user who needs access will have to be granted by the Administrator. When granting a user access to a case, as illustrated in appendices C and D, the blockchain is updated. Specifically, an entry called "*enabledAddress"* is created, which associates a user address with access to the case address.

5.5 Reporting

Figures 5.8, 5.9 and 5.10 below show individual logs from specific transactions. For instance, when a case is created, it is assigned a blockchain address. Fig 5.11 shows a blockchain log for adding a case with address *0xA8C633F61b6492ab3336Ed8BD51171a6f078cF0B*) assigned.

Figure 5.8: Case Addition Blockchain Log

When adding evidence, as shown in Figure 5.11, we observe that the same address that was used when creating the case (0xa8c633f61b6492ab3336ed8bd51171a6f078cf0b) is updated.

Figure 5.9: Add Evidence Log

A single reference can be used to display events associated with a case. This can be achieved by navigating to a case address like case 0xA8C633F61b6492ab3336Ed8BD51171a6f078cF0B. Figures 5.10 and 5.11 show transactions about a case which provide a chain of events about the case. Internal transactions show creation of a case or other case update.

The transactions section shows events associated with the case address such as adding evidence and enabling users with access. A report can therefore be generated about a case showing any interactions made including but not limited to creating of the case, adding, and updating of evidence, enabling, and removing user access, and events about users accessing the evidence.

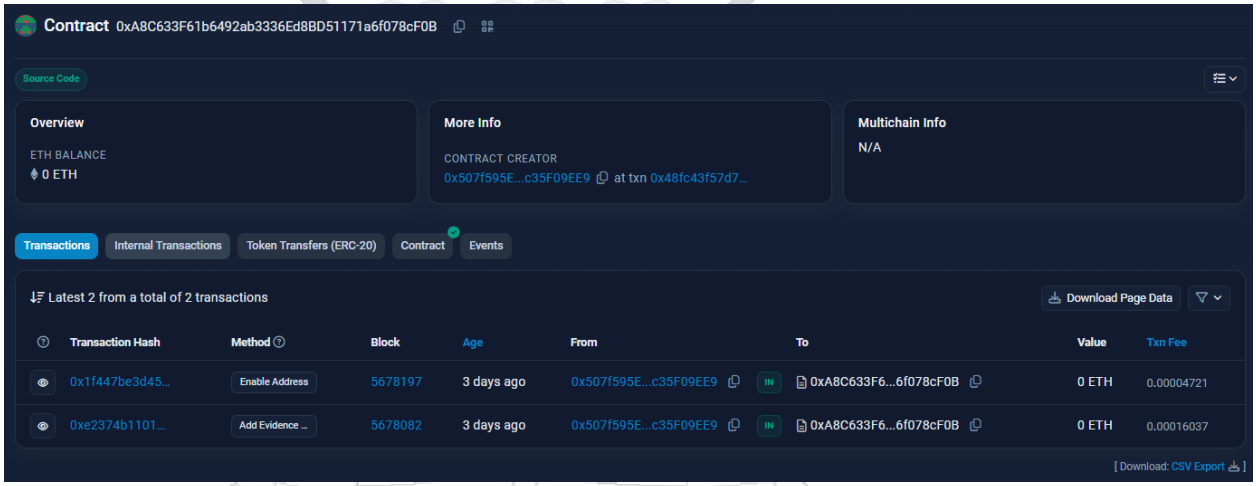


Figure 5.10: Case Transactions

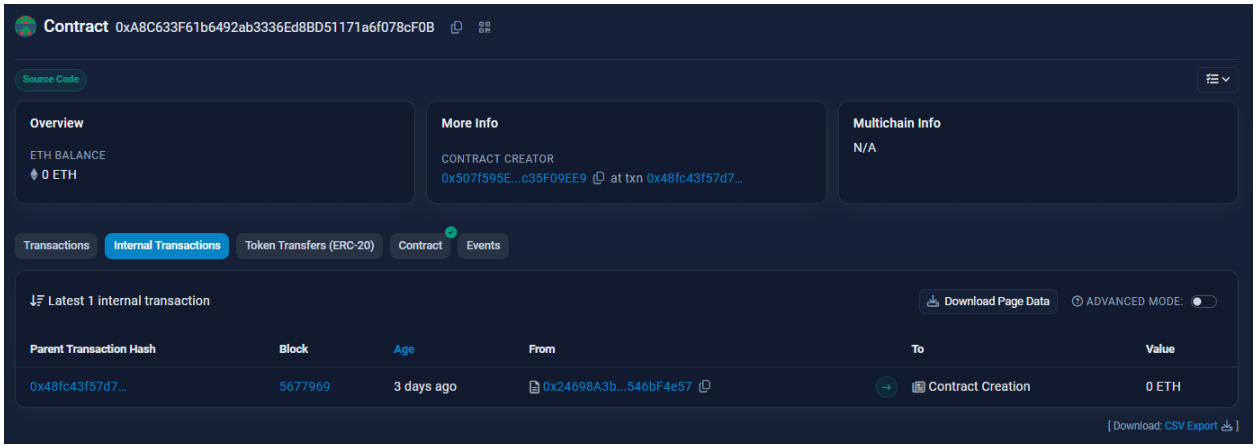


Figure 5.11: Case internal transactions Log.

Reports can also be generated about a user by checking the user address. Fig 5.12 below shows transactions associated to an admin user therefore showing chronological events of their actions on the system.

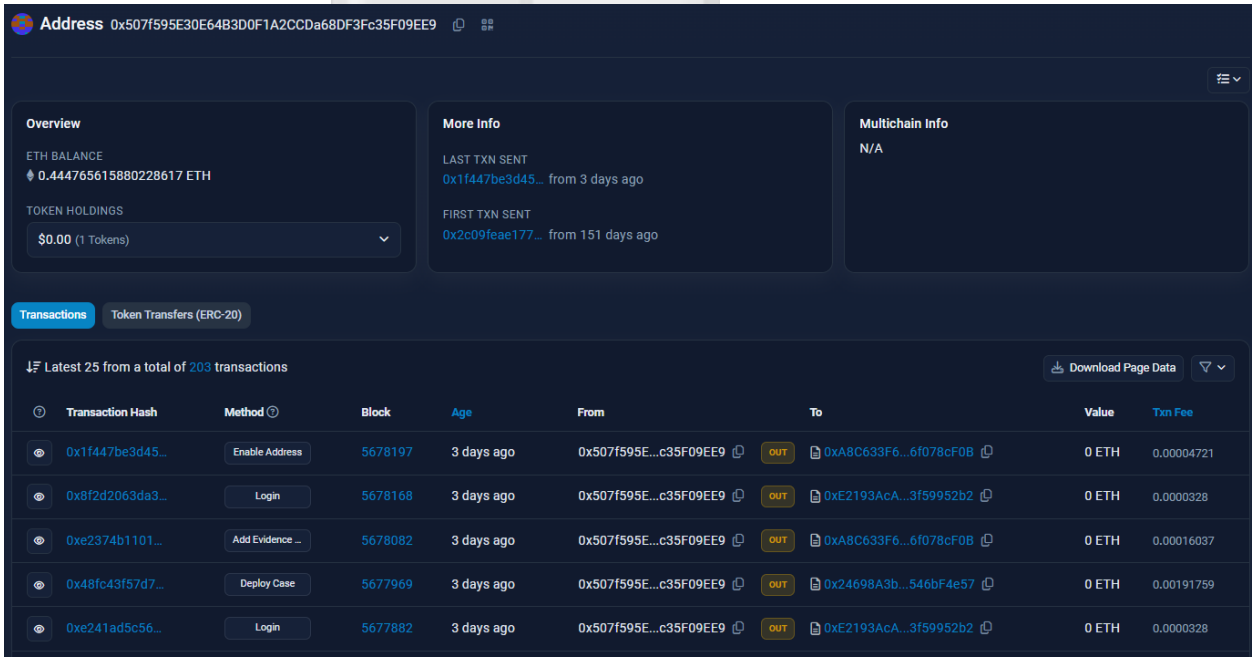


Figure 5.12: User transaction logs

5.6 Tests Results

To evaluate the key aspects of the study, the following tests were conducted:

5.6.1 Functionality Tests

Functionality tests are the tests that ascertain whether the system performs the tasks it was intended to do. Functional requirements provide a guide on what to test. To test Documentation of the evidence chain of custody, a case was created, evidence added, and specific users given access. All events from login of users, creation and updating of the case and evidence were found to have been stored on the blockchain. The security of the chain of custody was tested by trying to update evidence details and the result was a mere update to the blockchain thus keeping all other previous records intact. Users could only see cases they were assigned to thus passing the test on Access control of the system. The immutability of transaction meant that the reports and logs retrieved from the blockchain were secure.

5.6.2 Usability Tests

The solution was implemented on a web-based platform with a user-friendly interface, making it easy for users to input data and interact with the model effectively.

5.6.3 Compatibility Tests

Compatibility testing evaluates a system's ability to function effectively across diverse environments, including various web browsers, operating systems, devices, and other hardware. In this study, the initial compatibility test focused on assessing how well the web application performed on different browsers. We specifically focused on four major browsers: Firefox (version 124.0.2, 64-bit), Microsoft Edge (Version 123.0.2420.97, 64-bit), Chrome (Version 123.0.6312.122, 32-bit), and Safari (MacOS 14.4.). The primary objective was to determine whether the system seamlessly supported these selected browsers. All the tested browsers successfully displayed the web application and allowed seamless interaction. The NodeJS server was tested on Windows and Linux environments and worked efficiently for both operating systems.

5.6.4 Unit Tests

Unit testing was done throughout the development phase and across all modules. The testing focused on individual system components, and the results were consistently successful. Additionally, whenever a new feature or addition was introduced to the system, thorough testing ensured that it seamlessly integrated without disrupting the existing and completed sections.

5.6.5 Integration Tests

Integration tests were conducted to verify the efficient functioning of all system components. Each individual component underwent its own integration tests as part of unit testing. Once all modules were completed, comprehensive system integration tests were performed. These tests covered both local environments and live deployments. During testing, the web application and NodeJS server were executed locally to assess communication between the two components. Subsequently, the NodeJS server was deployed and tested for reachability from the locally running web application. Finally, the web application was deployed and verified to function seamlessly with the server.

5.7 Validation

5.7.1 Objective

The validation exercise aimed to assess the system's accuracy, reliability, and security using a structured methodology with quantitative and qualitative metrics to validate its functions, security, and performance. The functional testing the objective was to verify that the system correctly records, tracks, and updates digital evidence with metrics such as transaction logging, timestamp accuracy, and access control mechanisms. The objective for performance testing was to measure system efficiency with metrics such as transaction speeds. Finally, the objective for security and integrity testing was to validate that blockchain records are tamper-proof and resistant to unauthorized modifications.

5.7.2 Methodology

The validation was performed by simulating an investigative process that involved cases, evidence, users, and interactions throughout the investigation. The system's ability to handle evidence registration, access control, and evidence transfer was tested. The methodology included:

1. **Functional Testing**

    a. **Transaction Logging and Timestamp Verification**: Each entry was automatically assigned a timestamp, providing a clear record of when events such as access or modifications occurred. When a case or evidence was added, a corresponding blockchain entry was created, ensuring a reliable event log. Figure 5.13 below illustrates snippets of logged activities based on various actions carried out in the system. When an activity such as a case is added, a blockchain entry is created. Adding evidence also creates an entry. The system is thus ascertained to have kept the log of events. A closer look into a logged event shows details about an entry. These details include a timestamp that is a global standard that can be converted into any local time based on one's geographical location. This is useful in tracing when an event occurred and thus establishing a timeline.

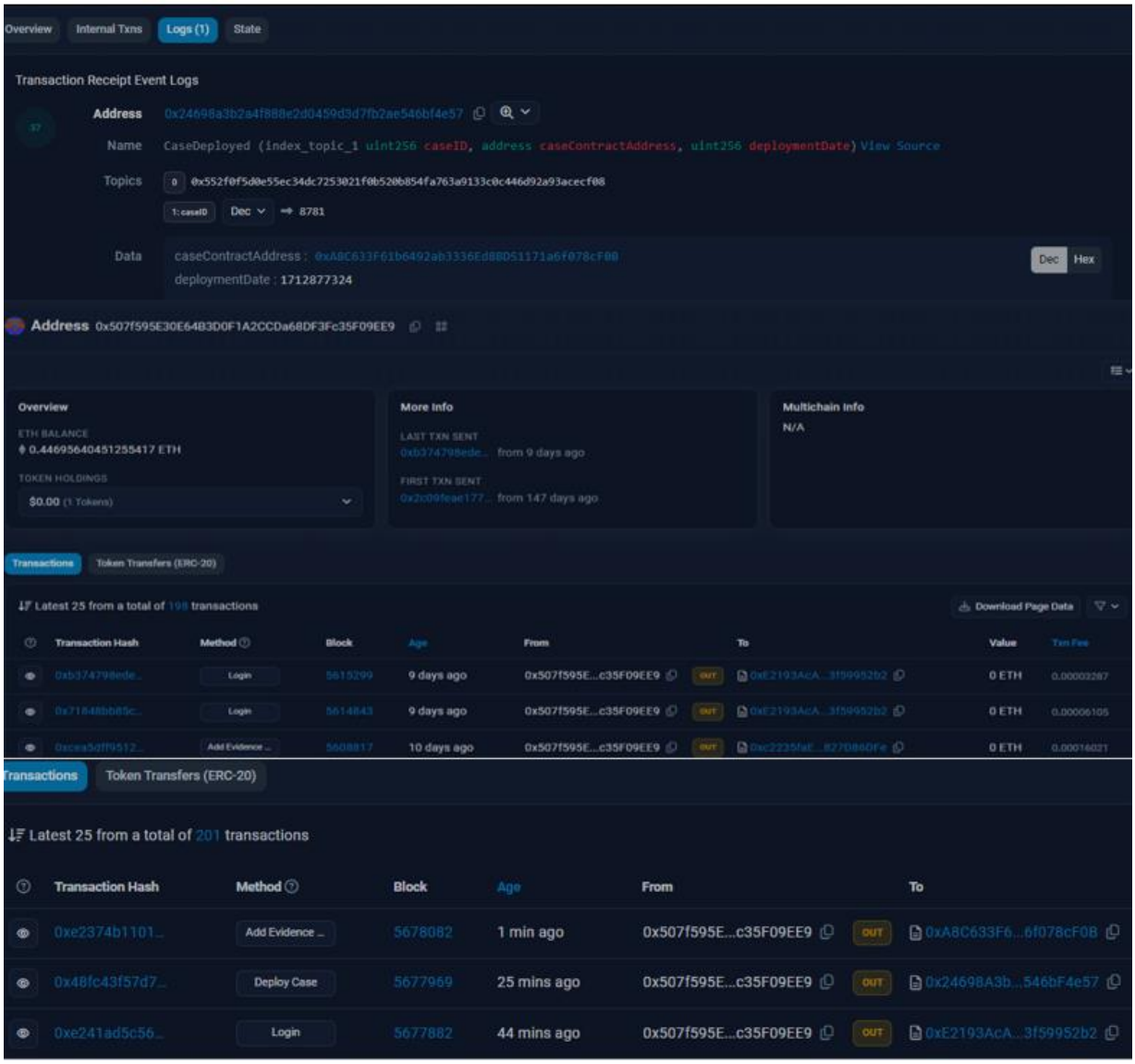


Figure 5.13: Transaction logging and time stamps

b. User Roles and Access Control: The system was designed to restrict privileges based on the role one is assigned. It also restricted access to only data that one had been allowed to such as a case. Only the system administrator was allowed to add and modify data such as cases and evidence in this prototype.

Any other user would only then be allowed to just view the cases and evidence. Figure 5.14 shows entries of a user being given access to a case (*Enable Address* method).

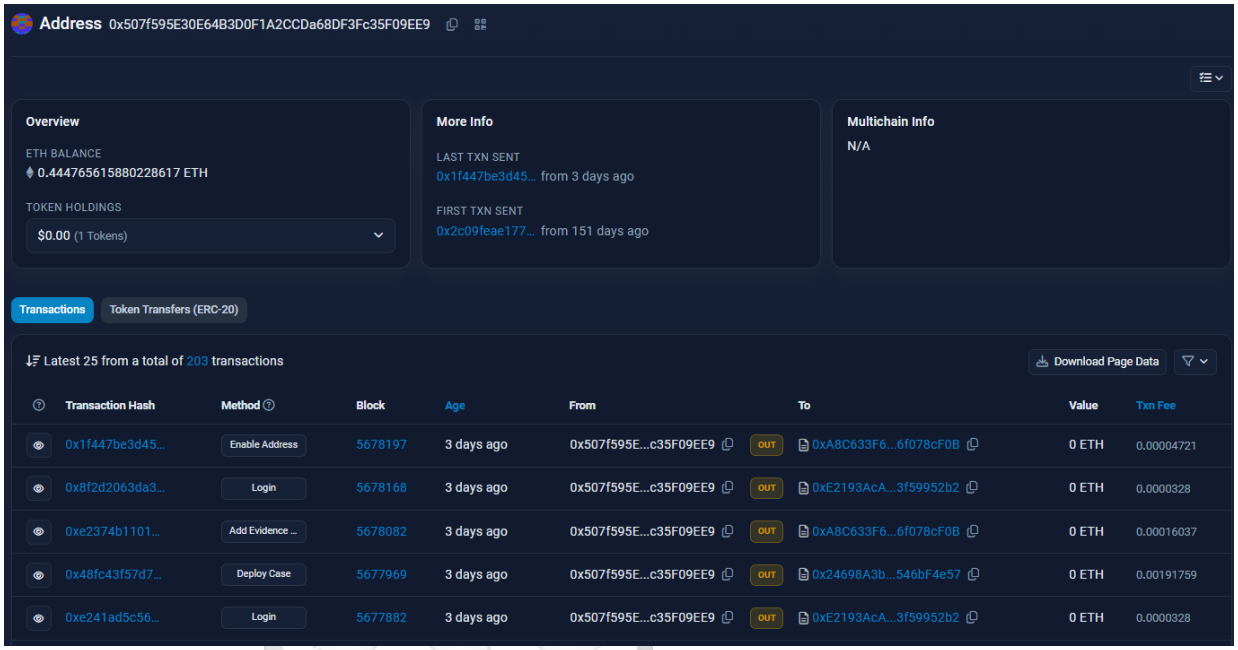


Figure 5.14: Access Control

2. **Data Integrity and Immutability Testing**: Data integrity and immutability were validated by the recording every event as a blockchain transaction, preserving original records and enabling a verifiable chain of custody. As seen from the above snippets, every event is recorded as a transaction on the blockchain. Modifications would therefore be recorded as new entries preserving the original data in earlier blockchain entries. This validates the integrity of data that was stored as the original copy will always be there. Any access or modification being recorded as an entry, also allows the chain of custody to be auditable and verified upon need.

3. **Performance Testing:** Performance and reliability were evaluated through API response time measurements and system uptime monitoring. Performance tests demonstrated high system reliability, achieving 99.87% uptime. API response times for case creation, evidence upload, and authentication were measured. Case creation was noted to be slightly slower than retrieval due to the longer time required to write data on the blockchain.

    a. **API Response:** API response time measured under normal and high-load conditions. (Figures rounded off due to variations of the network)

    Table 5.1: API Response times

API Endpoint	Operation	Avg Response Time (ms)	Peak Load Response Time (ms)
POST /cases	Create a new case	1200	1800
POST /evidence	Upload evidence	800	1350
GET /cases/:id	Fetch case details	220	500
POST /auth/login	User authentication	1100	2200

    b. System Uptime and Reliability: Availability monitoring over 7-day testing period. (A longer testing period is needed to get the true value under normal and peak conditions)

    Table 5.2: System Uptime

Metric	Value
Total Uptime	99.87%
Downtime	13 minutes
Successful Requests	99.5%
Failed Requests	0.5%

5.7.3 Conclusion

The validation exercise aimed to assess the system's accuracy, reliability, and security using a structured methodology with quantitative and qualitative metrics to validate its functions, security, and performance. The functional testing the objective was to verify that the system correctly records, tracks, and updates digital evidence with metrics such as transaction logging, timestamp accuracy, and access control mechanisms. The objective for performance testing was to measure system efficiency with metrics such as transaction speeds. Finally, the objective for security and integrity testing was to validate that blockchain records are tamper-proof and resistant to unauthorized modifications.

The validation was performed by simulating an investigative process that involved cases, evidence, users, and interactions throughout the investigation. The system’s ability to handle evidence registration, access control, evidence transfer was tested. Functional accuracy was validated through transaction logging and timestamp verification. Each entry was automatically assigned a timestamp, providing a clear record of when events such as access or modifications occurred. When a case or evidence was added, a corresponding blockchain entry was created ensuring a reliable event log. User roles and access control mechanisms were enforced, restricting data modification privileges to administrators while allowing controlled case access to other users. Data integrity and immutability was validated by recording every event as a blockchain transaction, preserving original records and enabling a verifiable chain of custody. Performance and reliability were evaluated through API response time measurements and system uptime monitoring. Performance tests demonstrated high system reliability, achieving 99.87% uptime. API response times for case creation, evidence upload, and authentication were measured. Case creation was noted to be slightly slower than retrieval due to the longer time required to write data on the blockchain. The validation exercise confirmed the system’s ability to track digital evidence while maintaining a chain of custody.

Chapter 6: Discussion of Key Results

6.1 Overview

This chapter examines the study findings in relation to the established objectives, research questions, and scope, providing explanations on the main topics covered. The aim of this research is to create a blockchain based solution that safeguards the integrity of the digital forensics chain of custody.

6.2 Objective 1: Identify Common Challenges with Digital Evidence

The first objective of this research was to identify the common challenges that affect the integrity of digital evidence along the chain custody. The literature review revealed that the increase in cybercrime has elevated importance of digital evidence in cybercrime investigation, as it is key to the identification and successful prosecution of cybercrime perpetrators. However, for the evidence to be admissible in court a chain of custody documenting how evidence was gathered, transported, analysed, and presented thereby proving that evidence has not been altered or changed must be kept.

Maintaining the integrity of digital evidence with regards to chain of custody is challenging due to the life cycle of digital evidence and its unique features. Unlike physical evidence, digital evidence is vulnerable to tampering as it can be easily copied, altered, damaged, or destroyed making it susceptible to malicious alteration or accidental changes. Documentation of digital evidence is also complicated by its ease of access and duplication. Another challenge is the security of chain of custody documentation, considering that evidence can be transferred between parties. It is against this backdrop that a robust system becomes imperative to protect the integrity of the chain of custody for digital evidence to be admissible in court.

6.3 Objective 2: Existing Solutions for Preserving a Digital Chain of Custody

The second research objective was to review the existing solutions for preserving integrity in a digital chain custody. A review of literature identified direct and indirect attempts to solve the digital evidence integrity challenges associated with digital chain of custody. Indirect attempts delivered the theoretical knowledge on evidence handling, potential

chain of custody issues that could compromise evidence integrity and proposed frameworks to solve the challenges. Most papers following this approach however lacked the technical aspect on how to practically apply the proposed frameworks. Direct attempts proposed various methods on how technology that can be integrated to maintain the chain of custody.

Existing tools reviewed included XML schema approach, Digital Evidence Cabinets (DEC), smartcards, and the Digital Evidence Management Framework (DEMF). The XML schema approach preserves the hash value of evidence files but is less accurate compared to metadata extraction and vulnerable to database manipulation. DEC enhances evidence handling with secure communication but faces performance issues with large data volumes and compatibility challenges. Smartcards ensure secure signing and verification of evidence but are limited to RAW format images and struggle with integration. DEMF provides comprehensive evidence management with encryption but encounters scalability and interoperability challenges.

In summary, while existing solutions offer insights into various approaches of maintaining the integrity of digital evidence, they face significant limitations. Interoperability issues hinder data exchange, while scalability remains a challenge due to large volumes of data. Reliance on manual processes and centralized databases increase the risk of human error, data entry mistakes, and tampering. These challenges create the need for a more robust, scalable, and tamper-resistant approach to digital evidence management.

6.4 Objective 3: Design Blockchain-based Digital Chain of Custody Prototype

The third research objective was to design, develop, and test a blockchain-based digital forensic chain platform. This objective was achieved through a structured process involving design, agile development, and testing. The design phase focused on defining system requirements and selecting an architecture that ensures secure and tamper-proof evidence management. The development phase followed an agile methodology, allowing for iterative refinement based on evolving requirements. The platform was built using Ethereum blockchain technology, with a Node.js backend server and a React.js frontend, ensuring seamless communication between components and effective interaction with the

blockchain. In the testing phase, the platform's effectiveness was evaluated through simulated forensic scenarios, security assessments, and performance analysis. The investigative process was simulated using key elements such as cases, evidence, users, and interactions, with blockchain transaction records providing a transparent audit trail of all activities. Comprehensive testing—including functionality, usability, compatibility, unit, and integration tests—validated the platform's ability to maintain a secure chain of custody. The successful implementation and validation, demonstrated through immutable transaction logs, confirmed the achievement of this objective and showcased blockchain's feasibility in addressing chain of custody challenges.

6.5 Objective 4: Validate The Effectiveness of the Proposed Prototype

The fourth objective was to validate the effectiveness of the proposed prototype. This involved simulating an investigative process that included key elements such as a case, evidence, users, and interactions throughout the investigation. The simulation tested the platform's ability to maintain a secure chain of custody, ensuring that all interactions and transactions were immutably recorded on the blockchain. Functional tests verified that the tool performed all intended tasks correctly, including evidence registration, access control, evidence transfer, and report generation. The prototype underwent validation by examining the record of transactions on the blockchain, providing insight into all interactions and transactions related to the various actors involved. The immutability of transaction records was validated by comparing the current state of evidence with its original blockchain-stored hash, providing a transparent and verifiable audit trail for all evidence interactions. The validation confirmed that the system can be relied upon to verify the integrity of evidence, as once a transaction is added to a block and subsequently to the blockchain, it becomes unalterable. This proved that the tool successfully met the study's objective.

Chapter 7: Conclusions, Recommendations and Future Work

7.1 Conclusions

This study explored the use of blockchain technology to enhance the integrity of digital evidence in the chain of custody. It was conducted within the framework of the Design Science Research (DSR) methodology, which structured the research process from problem identification to artifact evaluation and communication of results.

Aligning with the Problem Identification and Motivation phase of DSR, the first objective aimed to identify common challenges affecting evidence integrity along the chain of custody. The study found that digital evidence is highly susceptible to modification, unauthorized access, and loss of traceability due to the lack of tamper-proof mechanisms in traditional chain of custody processes. Additionally, centralized storage solutions and manual handling introduce risks such as human error, data entry mistakes, and evidence manipulation, all of which compromise the reliability and ultimate admissibility of digital evidence.

The second objective aligned with the Objectives Definition phase of DSR. A literature review revealed several approaches, including XML schema, Digital Evidence Cabinets (DEC), smartcards, and Digital Evidence Management Frameworks (DEMF). While these methods provide valuable insights, they suffer from limitations such as scalability challenges, interoperability issues, and reliance on centralized databases, which remain vulnerable to tampering and inefficiencies. These findings highlighted the need for a more robust and transparent solution capable of ensuring evidence integrity throughout its lifecycle, thereby shaping the objective to design a robust solution to protect digital evidence across its lifecycle.

To address this need, the third objective focused on the Design, Development, and Demonstration phases of DSR. This was achieved through a structured approach involving system design, agile development, and testing. The proposed platform was built using Ethereum blockchain technology, integrating a Node.js backend and a React.js frontend to enable secure evidence tracking. The solution ensured that each transaction and

interaction within the chain of custody was immutably recorded on the blockchain, providing a tamper-proof and transparent audit trail. The testing phase simulated forensic scenarios, demonstrating the platform's ability to maintain a secure and verifiable chain of custody.

The final objective was to validate the effectiveness of the proposed prototype, aligning with the Evaluation stage of DSR. This was accomplished through extensive testing, including functionality, usability, security, and performance assessments. The validation process confirmed that the blockchain-based tool successfully safeguarded the integrity of digital evidence by creating immutable transaction records and reducing risks associated with evidence manipulation. The decentralized nature of blockchain further minimized single points of failure, ensuring reliability and transparency in evidence management.

Finally, the results and insights were documented and presented as part of the research deliverables, fulfilling the Communication phase of the DSR methodology. In conclusion, this research demonstrated the feasibility of leveraging blockchain technology to address the challenges of digital evidence integrity in the chain of custody.

7.2 Future Work

Future work on this project involves developing a comprehensive framework that seamlessly integrates blockchain technology into digital evidence handling protocols. The framework could address both legal and technical considerations, offering standardized guidelines to facilitate the effective adoption of blockchain-based custody systems. Implementation of the framework will enhance the auditability, interoperability, and legal admissibility of digital evidence, while reducing risks associated with tampering, human error, and internal manipulation.

References

Akbarfam, A. J., Dorai, G., & Maleki, H. (2024, October). Secure Cross-Chain Provenance for Digital Forensics Collaboration. *In 2024 IEEE 6th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)* (pp. 389-398). IEEE.

Akbarfam, A. J., Heidaripour, M., Maleki, H., Dorai, G., & Agrawal, G. (2023, November). Forensiblock: A provenance-driven blockchain framework for data forensics and auditability. In *2023 5th IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)* (pp. 136-145). IEEE.

Alenezi, M. N., Alabdulrazzaq, H. K., & Mohammad, N. Q. (2022). Symmetric Encryption Algorithms: Review and Evaluation Study. *International Journal of Communication Networks and Information Security (IJCNIS)*, 12(2).. https://doi.org/10.17762/ijcnis.v12i2.4698 (Original work published August 23, 2020)

Alenezi, M. N., Alabdulrazzaq, H., & Mohammad, N. Q. (2020). Symmetric encryption algorithms: Review and evaluation study. *International Journal of Communication Networks and Information Security*, *12*(2), 256-272.

Al-Fedaghi, S. (2021). UML sequence diagram: an alternative model. *arXiv preprint arXiv:2105.15152.*

Ali, M., Ismail, A., Elgohary, H., Darwish, S., & Mesbah, S. (2022). A procedure for tracing chain of custody in digital image forensics: A paradigm based on grey hash and blockchain. *Symmetry*, *14*(2), 334.

Ali, V., Norman, A. A., & Azzuhri, S. R. B. (2023). Characteristics of blockchain and its relationship with trust. *Ieee Access*, *11*, 15364-15374.

Alqahtany, S. S., & Syed, T. A. (2024). ForensicTransMonitor: a comprehensive blockchain approach to reinvent digital forensics and evidence management. *Information*, *15*(2), 109. https://doi.org/10.3390/info15020109

Alqahtany, S. S., & Syed, T. A. (2024). ForensicTransMonitor: a comprehensive blockchain approach to reinvent digital forensics and evidence management. *Information*, *15*(2), 109.

Alruwaili, F. F. (2021). Custodyblock: A distributed chain of custody evidence framework. *Information*, *12*(2), 88. https://doi.org/10.3390/info12020088

Al-Saqqa, S., Sawalha, S., & AbdelNabi, H. (2020). Agile software development: Methodologies and trends. *International Journal of Interactive Mobile Technologies*, *14*(11).

Ammann, P., & Offutt, J. (2016). *Introduction to software testing*. Cambridge University Press.

Avison, D., & Fitzgerald, G. (2006). *Information systems development: Methodologies, techniques & tools* (4th ed.). McGraw-Hill Education.

Baskar, K. (2025). Emerging trends in digital forensics: Investigating cybercrime. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, *11*, 3645–3652. https://doi.org/10.32628/CSEIT251451

Bates, A., Pohly, D. J., & Butler, K. R. (2016). Secure and trustworthy provenance collection for digital forensics. *Digital Fingerprinting*, 141-176.

Bezuidenhout, R., Nel, W., & Maritz, J. M. (2023). Permissionless blockchain systems as pseudo-random number generators for decentralized consensus. *IEEE Access*, *11*, 14587-14611. https://ieeexplore.ieee.org/document/10042427

Booch, G. (2005). *The unified modeling language user guide*. Pearson Education India.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, *3*(2), 77-101.

Carrier, B., & Spafford, E. (2004). An event-based digital forensic investigation framework. *Digital Investigation*.

Casey, E. (2011). *Digital evidence and computer crime: Forensic science, computers, and the internet*. Academic press.

Chen, R., Wu, X., & Liu, X. (2023). Rsetp: A reliable security education and training platform based on the alliance blockchain. *Electronics*, *12*(6), 1427. https://doi.org/10.3390/electronics12061427

Ćosić, J., & Bacai, M. (2017). The Necessity of Developing a Standard for Exchanging a Chain of Custody of Digital Evidence Data A DEMF STORY. *International Journal of Computer Science and Information Security*, 15(11), 188–191.

Cosic, J., & Cosic, Z. (2012). Chain of custody and life cycle of digital evidence. *Computer technology and application*, *3*(2).

Ćosić, Z., Ćosić, J., & Bača, M. (2021). Chain of custody and life cycle of digital evidence. *Journal of Information and Organizational Sciences*, 45(1)

Fang, W., Chen, W., Zhang, W., Pei, J., Gao, W., & Wang, G. (2020). Digital signature scheme for information non-repudiation in blockchain: a state of the art review. *EURASIP Journal on Wireless Communications and Networking*, *2020*, 1-15.

Garfinkel, S. L. (2010). Digital forensics research: The next 10 years. *digital investigation*, *7*, S64-S73.

Geambasu, C. V., Jianu, I., Jianu, I., & Gavrila, A. (2011). Influence factors for the choice of a software development methodology. *Journal of Accounting and Management Information Systems (JAMIS)*, *10*(4), 479-494.

Giova, G. (2011). Improving chain of custody in forensic investigation of electronic digital systems. *International Journal of Computer Science and Network Security*, *11*(1), 1-9.

Haque, S., & Atkison, T. (2018). A forensic enabled data provenance model for public cloud. *Journal of Digital Forensics, Security and Law*, *13*(3), 7.

Hooda, I., & Chhillar, R. S. (2015). Software test process, testing types and techniques. *International Journal of Computer Applications*, *111*(13). https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-86.pdf

Igonor, O. S., Amin, M. B., & Garg, S. (2025). The Application of Blockchain Technology in the Field of Digital Forensics: A Literature Review. *Blockchains*, *3*(1), 5. https://doi.org/10.3390/blockchains3010005

International Organization for Standardization. (2012). *ISO/IEC 27037:2012 – Information technology – Security techniques – Guidelines for identification, collection, acquisition and preservation of digital evidence*. ISO.https://www.iso.org/standard/44381.html

International Organization for Standardization. (2016). *ISO 15489-1:2016 Information and documentation — Records management — Part 1: Concepts and principles*. ISO.ISO. https://www.iso.org/standard/62542.html

Jacobson, L., & Booch, J. R. G. (2021). The unified modeling language reference manual.

Karie, N. M., & Venter, H. S. (2015). Taxonomy of challenges for digital forensics. *Journal of forensic sciences*, *60*(4), 885-893.

Locard, E. (1930). The analysis of dust traces. *Am. J. Police Sci., 1*, 276.

Lone, A. H., & Mir, R. N. (2019). Forensic-chain: Blockchain based digital forensics chain of custody with PoC in Hyperledger Composer. *Digital investigation*, *28*, 44-55.

Malik, A., & Sharma, A. K. (2023). Blockchain-based digital chain of custody multimedia evidence preservation framework for internet-of-things. *Journal of Information Security and Applications*, *77*, 103579.

Naik, K., & Tripathy, P. (2011). *Software testing and quality assurance: theory and practice*. John Wiley & Sons.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

National Institute of Standards and Technology. (2006). *Guide to integrating forensic techniques into incident response* (Special Publication 800-86). U.S. Department of Commerce.

Object Management Group (OMG). (2017). OMG Unified Modeling Language (OMG UML), Superstructure, v2.5.1. OMG.

Pawar, R. P. (2015). A comparative study of agile software development methodology and traditional waterfall model. *IOSR Journal of Computer Engineering*, *2*(2), 1-8.

Pestana, G., Antunes, W., & Carvalho, J. (2023, November). Digital Chain of Custody Operational Framework. In *2023 IEEE International Workshop on Technologies for Defense and Security (TechDefense)* (pp. 417-422). IEEE.

Pourvahab, M., & Ekbatanifard, G. (2019). Digital forensics architecture for evidence collection and provenance preservation in iaas cloud environment using sdn and blockchain technology. *IEEE Access*, *7*, 153349-153364.

Prayudi, Y., & Sn, A. (2015). Digital chain of custody: State of the art. *International Journal of Computer Applications*, *114*(5).

Prayudi, Y., Ashari, A., & Priyambodo, T. K. (2014). Digital evidence cabinets: A proposed framework for handling digital chain of custody. *International Journal of Computer Applications*, *107*(9).

Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave macmillan.

PwC's Global Economic Crime and Fraud Survey 2022 https://www.pwc.com/gx/en/services/forensics/economic-crime-survey.html

Rasjid, Z. E., Soewito, B., Witjaksono, G., & Abdurachman, E. (2017). A review of collisions in cryptographic hash function used in digital forensic tools. *Procedia computer science*, *116*, 381-392.

Republic of Kenya. (2014). The Evidence Act (Cap. 80), Section 106B. National Council for Law Reporting. http://www.kenyalaw.org

Republic of Kenya. (2014). The Evidence Act (Cap. 80), Section 78A. National Council for Law Reporting. http://www.kenyalaw.org

Republic of Kenya. (2019). The Data Protection Act, 2019. Kenya Gazette Supplement No. 181 (Acts No. 24). Government Printer. https://ict.go.ke/data-protection-act-2019/

Riihiaho, S. (2018). Usability testing. *The Wiley handbook of human computer interaction*, *1*, 255-275.

Schatz, B., Mohay, G., & Clark, A. (2006). A correlation method for establishing provenance of timestamps in digital evidence. *digital investigation*, *3*, 98-107.

Shah, M. S. M. B., Saleem, S., & Zulqarnain, R. (2017). Protecting digital evidence integrity and preserving chain of custody. *Journal of Digital Forensics, Security and Law*, *12*(2), 12.

Sommerville, I. (2021). *Software engineering* (10th ed.). Pearson Education.

Spichiger, H., & Adelstein, F. (2025). Preserving meaning of evidence from evolving systems. *Forensic Science International: Digital Investigation*, *52*, 301867.

Stoykova, R. (2021). Digital evidence: Unaddressed threats to fairness and the presumption of innocence. *Computer Law & Security Review*, *42*, 105575.

Sultan, K., Ruhi, U., & Lakhani, R. (2018). Conceptualizing blockchains: characteristics & applications. *arXiv preprint arXiv:1806.03693*.

Valjarevic, A., & Venter, H. S. (2013, August). Implementation guidelines for a harmonised digital forensic investigation readiness process model. In *2013 Information Security for South Africa* (pp. 1-9). IEEE.

Vom Brocke, J., Hevner, A., & Maedche, A. (2020). Introduction to design science research. *Design science research. Cases*, 1-13. https://doi.org/10.1007/978-3-030-46781-4_1

Yeboah-Ofori, A., & Brown, A. D. (2020). Digital forensics investigation jurisprudence: issues of admissibility of digital evidence. *Journal of Forensic, Legal & Investigative Sciences*, *6*(1), 1-8.

Zarpala, L., & Casino, F. (2021). A blockchain-based forensic model for financial crime investigation: the embezzlement scenario. *Digital Finance*, *3*, 301-332. https://doi.org/10.1007/s42521-021-00035-5

Appendices

Appendix A: Similarity Report

turnitin

Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author:	Laban Nyarera
Assignment title:	Graduate Theses/Dissertations Submission and Similarity Che...
Submission title:	A Blockchain Based Tool for Enhancing Digital Evidence Integri...
File name:	30670_Laban_Nyarera_A_Blockchain_Based_Tool_for_Enhanci...
File size:	1.62M
Page count:	87
Word count:	16,116
Character count:	104,038
Submission date:	06-Apr-2025 06:00PM (UTC+0300)
Submission ID:	2636696674

A Blockchain Based Tool for Enhancing Digital Evidence Integrity in the Chain of Custody.pdf

ORIGINALITY REPORT

17%	19%	13%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	su-plus.strathmore.edu Internet Source	5%
2	www.mdpi.com Internet Source	1%
3	www.ijcnis.org Internet Source	1%
4	oa.upm.es Internet Source	1%
5	Auqib Hamid Lone, Roohie Naaz Mir. "Forensic-chain: Blockchain based digital forensics chain of custody with PoC in Hyperledger Composer", Digital Investigation, 2019 Publication	1%
6	www.ijcaonline.org Internet Source	1%
7	www.researchgate.net Internet Source	<1%
8	Submitted to Westcliff University Student Paper	<1%
9	papers.academic-conferences.org Internet Source	<1%
10	Submitted to University of Maryland, University College Student Paper	<1%

11	articles.forensicfocus.com Internet Source	<1%
12	www.coursehero.com Internet Source	<1%
13	scmsgroup.org Internet Source	<1%
14	www.ijrte.org Internet Source	<1%
15	Submitted to University of Glasgow Student Paper	<1%
16	arxiv.org Internet Source	<1%
17	Submitted to De Montfort University Student Paper	<1%
18	Submitted to UNICAF Student Paper	<1%
19	Submitted to Ghana Technology University College Student Paper	<1%
20	journal.binus.ac.id Internet Source	<1%
21	jyx.jyu.fi Internet Source	<1%
22	ijsart.com Internet Source	<1%
23	Submitted to University of South Australia Student Paper	<1%
24	Submitted to Strathmore University Student Paper	<1%
25	ampri.res.in Internet Source	<1%

Appendix B: Ethical Clearance Release Letter

Strathmore UNIVERSITY

26th March 2025

Laban Machuki Nyarera

Student Number: 119887

laban.nyarera@strathmore.edu, labzmachuki@gmail.com

Dear Laban,

RE: A Blockchain-Based Tool for Enhancing Digital Evidence Integrity in the Chain of Custody

This is to inform you that the Office of Graduate Studies on Tuesday March 18th, 2025, received your request on email for exemption from Ethical Clearance for the above Thesis. However, it was noted that the Research Services Office and The Strathmore University Institutional Scientific and Ethical Review Committee (SU-ISERC) objected to reviewing your study since you have already collected data and written the Thesis. The scientific & ethical review/approval process is ONLY done before the commencement of any experiments, implementation or any collection of data (primary or secondary).

The office notes that: On the grounds of not having submitted your research proposal, with reason of ethical approval not being compulsory at the time of your research study in the University. This is a letter for you to proceed with the next steps of your academic requirements.

Please be advised, that in future, all research proposals should be submitted to the SU-ISERC through the RHInnO Ethics platform: https://strathmoreuniversity.rhinno.net/login

Disclaimer: 1) This is not in any way an ethical approval letter. 2) Should there be any legal implications/actions emanating from the research in terms of any ethical violations, you will be personally liable.

Yours sincerely,

Prof. Bernard Shibwabo
Director of Graduate Studies

Appendix C:NACOSTI Letter

REPUBLIC OF KENYA

NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Ref No: **423272**

Date of Issue: **15/April/2025**

RESEARCH LICENSE

This is to Certify that Mr.. LABAN machuki NYARERA of Strathmore University, has been licensed to conduct research as per the provision of the Science, Technology and Innovation Act, 2013 (Rev.2014) in Nairobi on the topic: A Blockchain-Based Tool for Enhancing Digital Evidence Integrity in the Chain of Custody for the period ending : 15/April/2026.

License No: **NACOSTI/P/25/418250**

423272

Applicant Identification Number

Director General
NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Verification QR Code

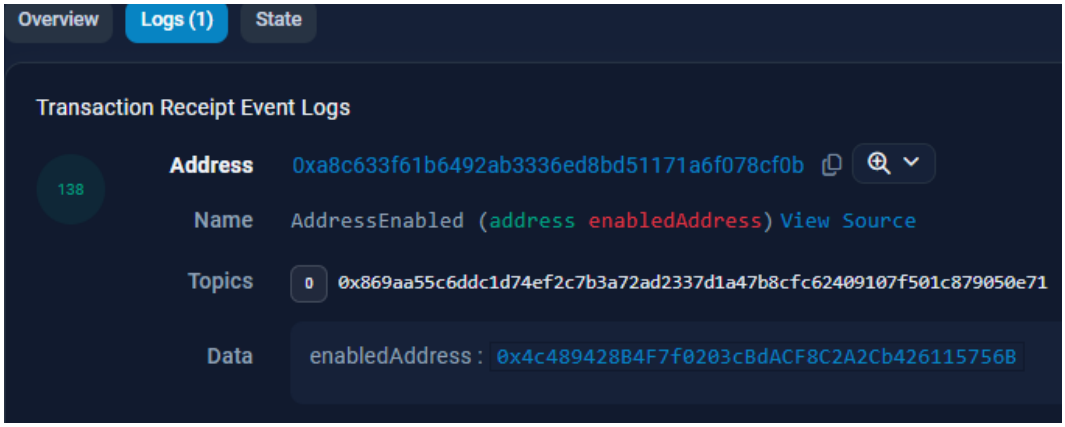
NOTE: This is a computer generated License. To verify the authenticity of this document, Scan the QR Code using QR scanner application.

See overleaf for conditions

Appendix D: Registration Function

```
// Registering a new user
router.post('/register', verifyToken, async (req, res) => {
  const { password } = req.body;

  try {
    user = await req.user

    // Ensure the caller is admin
    // if (user && Number(user.userID) === 0) {
    if (user && isAdmin(user)) {

      // Get the nextUserId from the accountsContract
      const nextUserId = await getNextUserId();

      // Using selected account for registration
      const userAddress = wallet.deriveChild(Number(nextUserId)).address.toLowerCase(); // Using the selected address

      // Securely hash the password
      const saltRounds = 10; // Number of salt rounds (adjust for your needs)
      bcrypt.hash(password, saltRounds, async (err, hash) => {
        if (err) {
          console.error(err);
          res.status(500).send('Error hashing password');
        }
        else {       You, 1 second ago • Uncommitted changes
          try {
            // Call the Ethereum smart contract to create a new user
            const tx = await accountsContract.connect(admin).addUser(userAddress, hash);
            //await tx.wait(1);
            res.status(201).send(`User ${nextUserId} registered successfully.`);

          } catch (error) {
            res.status(500).send({ message: error.revert ? error.reason : error.message });
          }

        }
      });
    } else {
      res.status(403).json({ message: 'Unauthorized - accounts access required' });
    }

  } catch (error) {
    console.log(error.message);
    res.status(500).send({ message: error.revert ? error.reason : error.message });
  }
});
```

Appendix E: Assigning Case to User Logs



Appendix F: Assigning Case to User